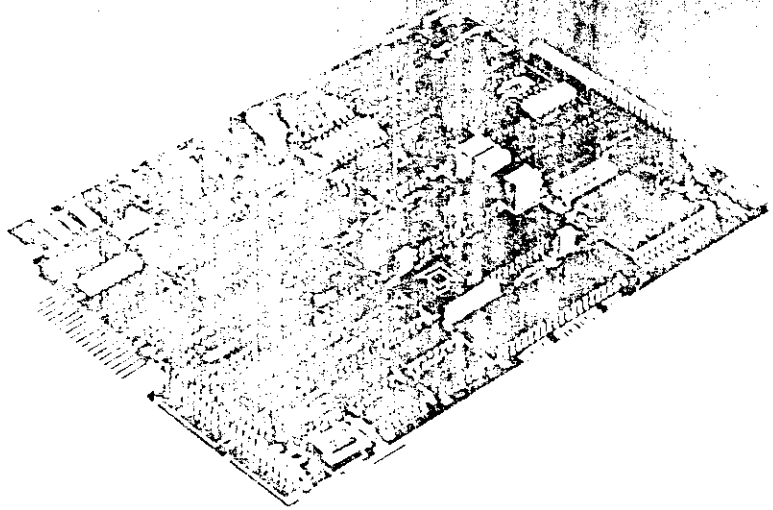
# XEBEC S1410

5.25 Inch Winchester Disk Controller



Owner's Manual


# XEBEC S1410

5.25 Inch Winchester Disk Controller

Owner's Manual



Document # [illegible]

REVISION [illegible]

April [illegible]



© XEBEC SYSTEMS INCORPORATED 1982

# CONTENTS

## INTRODUCTION


| | | Page |
|---|---|---|
| 1.1 | GENERAL | 1 |
| 1.2 | DESCRIPTION | 1 |
| 1.3 | FUNCTIONAL ORGANIZATION | 1 |
| 1.3.1 | Host Interface | 1 |
| 1.3.2 | Processor | 1 |
| 1.3.3 | State Machine | 1 |
| 1.3.4 | SERDES | 2 |
| 1.3.5 | Data Separator | 2 |
| 1.3.6 | Sector Buffer | 2 |


## SPECIFICATIONS


| | | |
|---|---|---|
| 2.1 | GENERAL | 5 |
| 2.2 | ELECTRICAL | 5 |
| 2.3 | PHYSICAL SPECIFICATIONS | 5 |
| 2.4 | ENVIRONMENTAL REQUIREMENTS | 5 |
| 2.5 | CONNECTORS | 5 |
| 2.6 | CONNECTOR PIN ASSIGNMENTS | 7 |


## BOARD SETUP


| | | |
|---|---|---|
| 3.1 | GENERAL | 9 |
| 3.2 | BOARD SETUP | 9 |
| 3.3 | MOUNTING CONTROLLER | 9 |
| 3.4 | CONNECTING CABLES | 9 |
| 3.5 | MULTIPLE CONTROLLERS | 9 |
| 3.6 | ADDRESS JUMPER GROUP | 12 |


## THEORY OF OPERATION


| | | |
|---|---|---|
| 4.1 | GENERAL | 13 |
| 4.1.1 | Conventions | 13 |
| 4.1.2 | Name and Abbreviations | 13 |
| 4.1.3 | Signal Definitions | 13 |
| 4.2 | BASIC OPERATING CONFIGURATION | 15 |
| 4.3 | DETAILED DESCRIPTION | 15 |
| 4.3.1 | Controller Selection | 15 |
| 4.3.2 | Command Mode | 16 |
| 4.3.3 | Data Transfer | 17 |
| 4.3.4 | Status Bytes | 17 |
| 4.4 | PROGRAMMING INFORMATION | 18 |
| 4.5 | COMMANDS | 18 |
| 4.5.1 | Control Byte | 19 |
| 4.5.2 | Logical Address (High, Middle, and Low) | 20 |
| 4.5.3 | Test Drive Ready (Class 0, Opcode 00) | 20 |
| 4.5.4 | Recalibrate (Class 0, Opcode 01) | 20 |
| 4.5.5 | Reserved (Class 0, Opcode 02) | 21 |
| 4.5.6 | Request Sense Status (Class 0, Opcode 03) | 21 |
| 4.5.7 | Format Drive (Class 0, Opcode 04) | 23 |
| 4.5.8 | Check Track Format (Class 0, Opcode 05) | 24 |
| 4.5.9 | Format Track (Class 0, Opcode 06) | 24 |
| 4.5.10 | Format Bad Track (Class 0, Opcode 07) | 25 |
| 4.5.11 | Read (Class 0, Opcode 08) | 25 |

| | | Page |
|---|---|---|
| 4.5.12 | Reserved (Class 0, Opcode 09) | 26 |
| 4.5.13 | Write (Class 0, Opcode 0A) | 26 |
| 4.5.14 | Seek (Class 0, Opcode 0B) | 26 |
| 4.5.15 | Initialize Drive Characteristics (Class 0, Opcode 0C) | 27 |
| 4.5.16 | Read ECC Burst Error Length (Class 0, Opcode 0D) | 28 |
| 4.5.17 | RAM Diagnostic (Class 7, Opcode 00) | 28 |
| 4.5.18 | Reserved (Class 7, Opcode 01) | 28 |
| 4.5.19 | Reserved (Class 7, Opcode 02) | 28 |
| 4.5.20 | Drive Diagnostic (Class 7, Opcode 03) | 29 |
| 4.5.21 | Controller Internal Diagnostics (Class 7, Opcode 04) | 29 |
| 4.5.22 | Read Long (Class 7, Opcode 05) | 29 |
| 4.5.23 | Write Long (Class 7, Opcode 06) | 30 |
| 4.6 | SECTOR FORMAT | 30 |
| 4.7 | EXECUTION OF DIAGNOSTICS | 31 |
| 4.8 | ERROR CORRECTION PHILOSOPHY | 31 |
| 4.9 | ALTERNATE TRACK ASSIGNMENT AND HANDLING | 32 |
| 4.10 | SECTOR INTERLEAVING | 33 |
| | **APPENDIX A** CONFIGURATION PARAMETERS FOR VARIOUS DISK DRIVES | A-1 |
| | **APPENDIX B** SAMPLE HARDWARE INTERFACE AND PROGRAMMING EXAMPLES FOR A Z-80 BASED MICROPROCESSOR TO INTERFACE TO THE XEBEC S1410 CONTROLLER | B-1 |

## LIST OF FIGURES

| Figure | Description | Page |
|---|---|---|
| 1-1 | Picture of S1410 Disk Controller | 2 |
| 1-2 | S1410 controller functional organization | 3 |
| 2-1 | Controller board dimensions | 6 |
| 3-1 | Cable, connector and jumper locations | 10 |
| 3-2 | Operating setups | 11 |
| 3-3 | Normal (factory-installed) address jumper | 12 |
| 3-4 | Changed address jumper (controller 1) | 12 |
| 4-1 | Basic operating configuration | 16 |
| 4-2 | Controller select timing | 16 |
| 4-3 | Data transfer to host timing | 17 |
| 4-4 | Data transfer from host timing | 18 |
| 4-5 | Device control block (DCB), format | 18 |
| 4-5a | Completion status bytes | 19 |
| 4-6 | Sector format | 31 |
| 4-10 | Track format example of 32 sectors-per-track with an interleave factor of 5 | 33 |



## LIST OF TABLES

| Table | Description | Page |
|---|---|---|
| 2-1 | Controller electrical requirements | 5 |
| 2-2 | Controller board specifications (physical) | 5 |
| 2-3 | Controller environmental limits | 5 |
| 2-4 | Controller mating connectors | 7 |
| 2-5 | Connector J1, control signals, pin assignments | 7 |
| 2-6 | Connectors J2 and J3, data signals, pin assignments | 7 |
| 2-7 | Connector P2, host interface, pin assignments | 8 |
| 2-8 | Connector P1, power supply, pin assignments | 8 |
| 3-1 | Jumper locations (W1, W2 and W3) | 9 |
| 4-1 | Host bus status signals | 13 |
| 4-2 | Summary of host bus status signals | 14 |
| 4-3 | Controller-host handshaking | 14 |
| 4-4 | Host bus control signals | 14 |
| 4-5 | Host bus data signals | 15 |
| 4-6 | Type 0 error codes, disk drive | 22 |
| 4-7 | Type 1 error codes, controller | 22 |
| 4-8 | Types 2 and 3 error codes, command and miscellaneous | 23 |
| 4-9 | Sector field description | 30 |

## COMMAND CODE SUMMARY

| Command Code (hex) | Meaning | Page |
|---|---|---|
| 00 | Test for drive ready | 20 |
| 01 | Recalibrate (position heads on track 00) | 20 |
| 02 | Reserved (not used) | 21 |
| 03 | Request sense status | 21 |
| 04 | Format disk drive | 23 |
| 05 | Check track format | 24 |
| 06 | Format track | 24 |
| 07 | Format bad track | 25 |
| 08 | Read | 25 |
| 09 | Reserved (not used) | 26 |
| 0A | Write | 26 |
| 0B | Seek | 26 |
| 0C | Initialize drive characteristics | 27 |
| 0D | Read ECC burst error length | 28 |
| E0 | Ram diagnostic (sector buffer test) | 28 |
| E1-E2 | Reserved (not used) | 28 |
| E3 | Drive diagnostic | 29 |
| E4 | Controller internal diagnostics | 29 |
| E5 | Read long | 29 |
| E6 | Write long | 30 |

# CHAPTER 1
# INTRODUCTION

## 1.1 GENERAL

The Xebec S1410 Disk Controller can control the operation of up to two 5 1/4-inch Winchester disk drives that have interfaces that are compatible with the interface of the ST-506 disk drive made by Seagate Technology. This means that the S1410 Controller can operate with a large and growing class of 5 1/4-inch Winchester disk drives.

## 1.2 DESCRIPTION

The S1410 Controller, shown in Figure 1-1, is packaged on a compact printed circuit board whose dimensions are 5-3/4 by 8 inches. The board with this popular form factor mounts easily on many 5 1/4-inch drives. If not mounted directly on the drive, the controller takes up very little space in a typical drive enclosure. Because the controller uses the Shugart Associates System Interface (SASI), it does not require special or complex design considerations in order to communicate with popular host buses. The following list highlights the operating and design features of the controller.

- Interlocked data transfer through the Shugart Associates System Interface (SASI).
- Microprocessor-based architecture (patent pending).
- Full-sector buffer (256 or 512 bytes).
- Hardware 32-bit ECC polynomial with 11-bit burst correction.
- Field-proven data separator.
- Seagate Technology disk interface.
- Automatic retries during disk access.
- Internal Diagnostics.
- Automatic burst error detection and correction.
- Separate sector format for ID and data fields with individual ECC fields for both the ID and data fields.
- High level command set.
- Variable Interleave.

## 1.3 FUNCTIONAL ORGANIZATION

The simplified block diagram in Figure 1-2 shows the functional organization of the Controller. Only the major areas are shown.

### 1.3.1 Host Interface

The host interface connects the internal data bus to the host adapter; the state machine controls the movement of data and commands through the host interface.

### 1.3.2 Processor

The eight-bit processor is the intelligence of the controller; it monitors and controls the operation of the controller.

### 1.3.3 State Machine

The state machine controls and synchronizes the operation of the host adapter, SERDES, and sector buffer.

### 1.3.4 SERDES

The serializer/deserializer (SERDES) converts parallel data from the internal data bus to serial data for transfer to a selected disk drive. It converts serial data from the selected disk drive to parallel data which it places on the internal data bus.

### 1.3.5 Data Separator

The data separator converts serial NRZ data to MFM for transfer to the selected disk drive. It converts MFM data coming from the selected disk drive to serial NRZ data for the SERDES.

### 1.3.6 Sector Buffer

The sector buffer stages data transfers between the disk and the host to prevent data overruns.

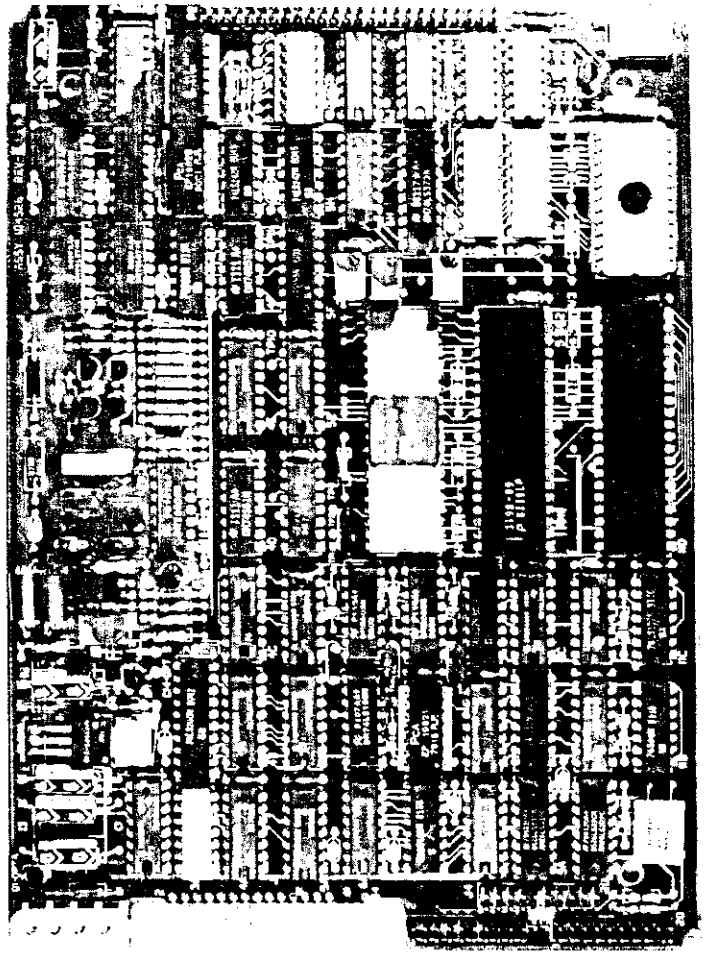


*FIGURE 1-1 S1410 DISK CONTROLLER*



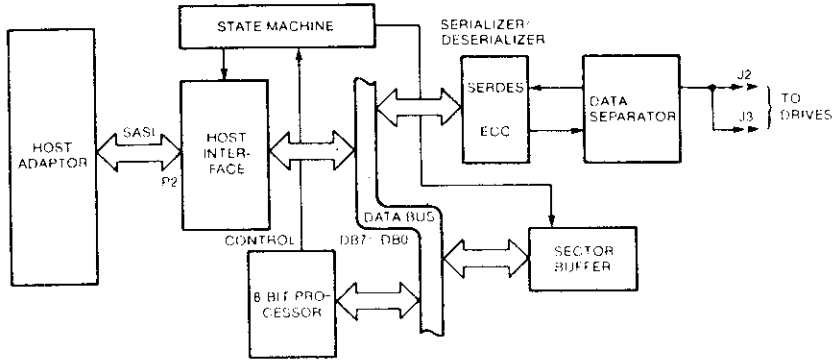



*FIGURE 1-2 S1410 CONTROLLER, FUNCTIONAL ORGANIZATION*

# CHAPTER 2
# SPECIFICATIONS

## 2.1 GENERAL

This chapter contains the overall specifications for the Controller. These specifications are meant to guide the user in placing the controller into operation. Some of the specifications indicate limits; the user must adhere to these in order to operate the controller successfully.

## 2.2 ELECTRICAL

Table 2-1 lists the electrical requirements of the controller.

***TABLE 2-1 CONTROLLER ELECTRICAL REQUIREMENTS***

**NOTE:** All measurements are made on the controller printed circuit board at the power connector P1.

| Voltage | Range | Current |
|---|---|---|
| +5.0 Vdc | 4.75 to 5.25 Vdc | 2.5 Amp. Max.<br>2.0 Amp. Typ. |
| +12.0 Vdc | 10.8 to 13.2 Vdc | 66.0 ma. Max.<br>48.0 ma. Typ. |

**NOTE:** The maximum conducted power supply ripple must not exceed 0.10 volts rms, from 0.1 to 25 MHz.

## 2.3 PHYSICAL SPECIFICATIONS

Table 2-2 lists the specifications of the controller board and Figure 2-1 illustrates the dimensions of the board.

***TABLE 2-2 CONTROLLER BOARD SPECIFICATIONS***

| Item | Measurement |
|---|---|
| Width(W) | 5.75 inches |
| Length (L) | 8.00 inches |
| Height (H) (Board thickness, components and lead protrusion) | 0.75 inches |
| Weight | 9.0 ounces |

## 2.4 ENVIRONMENTAL REQUIREMENTS

The controller will operate under the environmental conditions listed in Table 2-3. The controller does not normally require fans in standard operating environments where airflow is not restricted.

***TABLE 2-3 ENVIRONMENTAL LIMITS***

| Item | Measurement |
|---|---|
| Temperature | 0 to 55 degrees Celsius |
| Relative Humidity | 10 to 95 percent non-condensing |
| Altitude | Sea level to 10,000 feet |

## 2.5 CONNECTORS

Table 2-4 lists the Controller mating connectors.

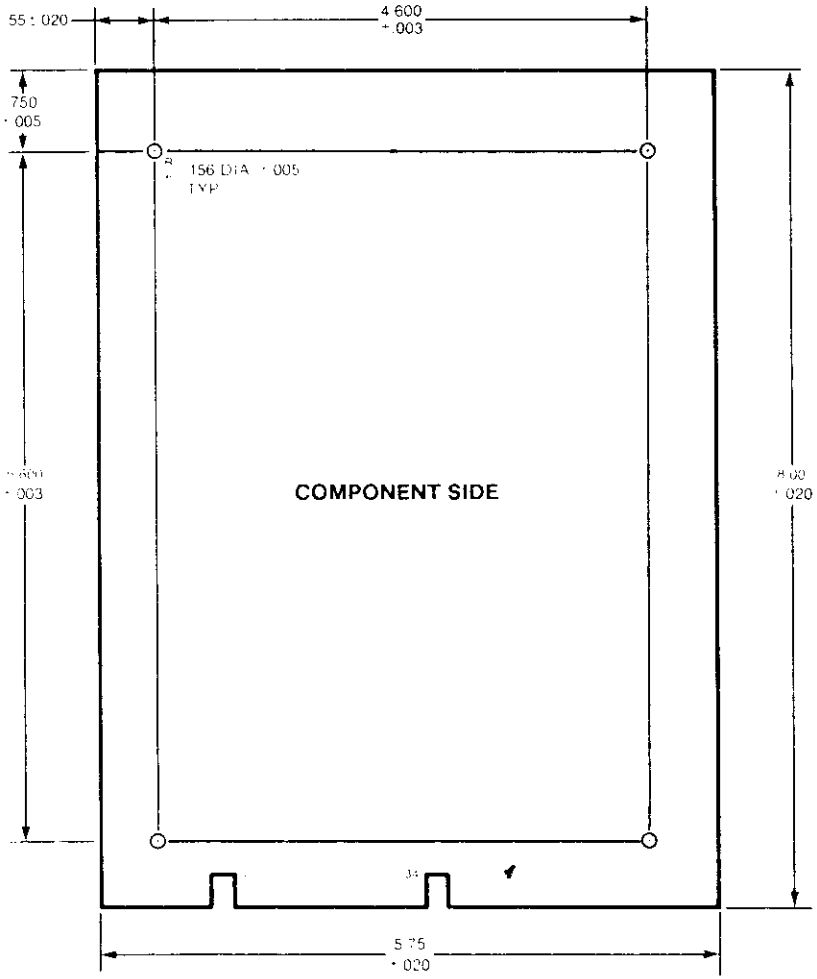



*FIGURE 2-1 CONTROLLER BOARD DIMENSIONS*



***TABLE 2-4 CONTROLLER MATING CONNECTORS***

| Designation | Function | Type/Source (or equivalent) |
|---|---|---|
| J1 | Drive control signals | AMP 88373-3 |
| J2,J3 | Drive data signals | AMP 86904-1 |
| J4 | Test connector | Not applicable |

**NOTE:** The user *must not* connect a cable to connector J4.

| | | |
|---|---|---|
| P1 | Power Supply | AMP 1-480424-0(housing)<br>AMP 350078-4(pins) |
| P2 | Host interface signals | AMP 86916-1 |

## 2.6 CONNECTOR PIN ASSIGNMENTS

Tables 2-5 through 2-8 list the pin assignments of the connectors on the controller board. The tables identify the signals on the pins. The connector P2 signals are defined in Chapter 4, Theory of Operation.

***TABLE 2-5***
***CONNECTOR J1, CONTROL SIGNALS, PIN ASSIGNMENTS***

| Signal Pin | Ground Return | Signal Name |
|---|---|---|
| 2 | 1 | Reduced Write Current |
| 4 | 3 | Head Select $2^2$ |
| 6 | 5 | Write Select Gate |
| 8 | 7 | Seek Complete |
| 10 | 9 | Track 00 |
| 12 | 11 | Write Fault |
| 14 | 13 | Head Select $2^0$ |
| 16 | 15 | Reserved |
| 18 | 17 | Head Select $2^1$ |
| 20 | 19 | Index |
| 22 | 21 | Ready |
| 24 | 23 | Step |
| 26 | 25 | Drive Select 1 |
| 28 | 27 | Drive Select 2 |
| 30 | 29 | Reserved |
| 32 | 31 | Reserved |
| 34 | 33 | Direction In |

***TABLE 2-6***
***CONNECTORS J2 and J3, DATA SIGNALS, PIN ASSIGNMENTS***

| Signal Pin | Ground Return | Signal Name |
|---|---|---|
| 1 | 2 | Drive Selected |
| 5 | 6 | Spare |
| 7 | 8 | Reserved |
| — | — | Spares, pins 9 and 10 |
| 11 | 12 | Ground (GND) |
| 13 | | MFM Write Data+ |
| 14 | | MFM Write Data- |
| 15 | 16 | Ground (GND) |
| 17 | | MFM Read Data+ |
| 18 | | MFM Read Data- |
| 19 | 20 | Ground (GND) |

*TABLE 2-7*
*CONNECTOR P2, HOST INTERFACE, PIN ASSIGNMENTS*

| Signal Pin | Ground Return | Signal Name |
|---|---|---|
| 2 | 1 | DATA0- |
| 4 | 3 | DATA1- |
| 6 | 5 | DATA2- |
| 8 | 7 | DATA3- |
| 10 | 9 | DATA4- |
| 12 | 11 | DATA5- |
| 14 | 13 | DATA6- |
| 16 | 15 | DATA7- |
| 18 | 17 | Spare |
| 20 | 19 | Spare |
| 22 | 21 | Spare |
| 24 | 23 | Spare |
| 26 | 25 | Spare |
| 28 | 27 | Spare |
| 30 | 29 | Spare |
| 32 | 31 | Spare |
| 34 | 33 | Spare |
| 36 | 35 | BUSY- |
| 38 | 37 | ACK- |
| 40 | 39 | RST- |
| 42 | 41 | MSG- |
| 44 | 43 | SEL- |
| 46 | 45 | C-/D |
| 48 | 47 | REQ- |
| 50 | 49 | I-/O |

*TABLE 2-8*
*CONNECTOR P1, POWER SUPPLY, PIN ASSIGNMENTS*

| Pin Number | Voltage |
|---|---|
| 1 | +12 Vdc |
| 2 | Ground return |
| 3 | Ground return |
| 4 | +5 Vdc |



# CHAPTER 3
# BOARD SETUP

## 3.1 GENERAL

This chapter contains the information for setting up and installing the controller before placing it in operation. These preparatory steps require the proper placement of jumpers, mounting the controller in its operating environment, and properly connecting the cables. In addition, the user has the option of using more than one controller with the host adapter in his system. Instructions for connecting multiple controllers appear later in the chapter.

## 3.2 BOARD SETUP

Setting up the Controller for operation requires checking that the factory-installed jumpers are according to the listing in Table 3-1. Figure 3-1 shows the locations of the jumpers on the Controller.

*TABLE 3-1 JUMPER LOCATIONS*

| Designation | Function | Connection and Result |
|---|---|---|
| W1 | Factory test | Must be installed |
| W2 | Factory test | Must be installed |
| W3 | Selects sector size | SS to 2: 256-byte sector, 32 Sectors/Track<br>SS to 5: 512-byte sector, 17 Sectors/Track |

## 3.3 MOUNTING CONTROLLER

The controller board has four mounting holes. It can be mounted anywhere within the drive enclosure so long as it receives airflow.

## 3.4 CONNECTING CABLES

Before the controller can be placed in operation, the cables to the drive and host must be connected. These cables are listed below:

| | |
|---|---|
| J1 | Control Cable (controller to last drive): maximum 20 feet |
| J2 | Data Cable: maximum 20 feet |
| J3 | Data Cable: maximum 20 feet (optional second drive) |
| P1 | Power Cable |
| P2 | Host Interface Cable: maximum 15 feet |

**NOTE:** Do not attempt to connect a cable to connector J4. Connector J4 is for factory test only.

Figure 3-1 shows the connector locations.

## 3.5 MULTIPLE CONTROLLERS

A separate Controller is required for each additional pair of drives. Figure 3-2 shows two operating setups; (A) using one Controller and (B) using two Controllers. Notice the terminator (resistor pack) in both drawings. The terminator is at position 5J on the board; when multiple Controllers are used, the terminator must be installed only in the last board in the daisy chain.

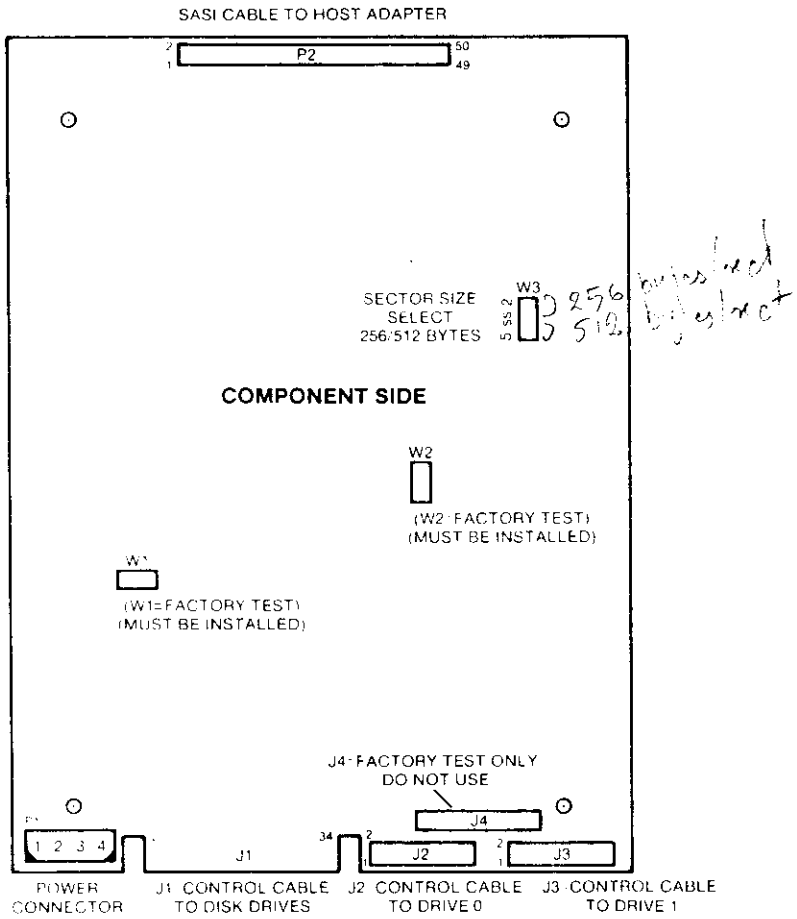



***FIGURE 3-1 CABLE, CONNECTOR, AND JUMPER LOCATIONS***



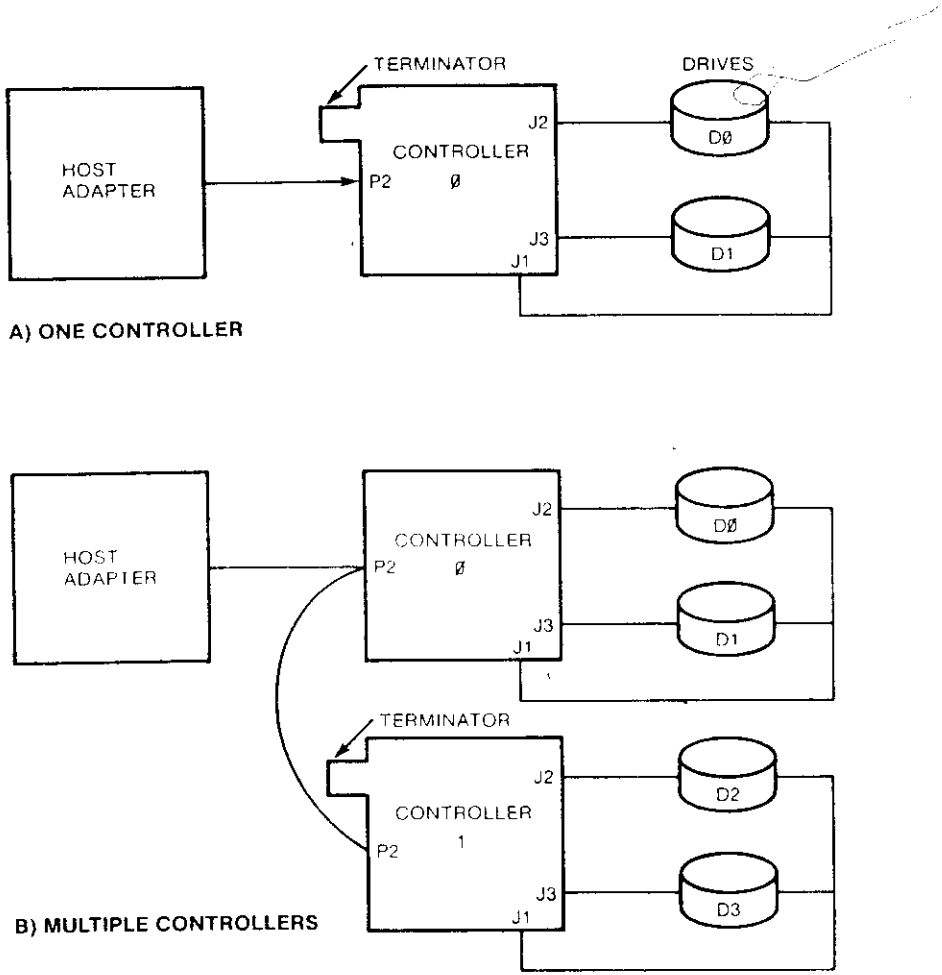



***FIGURE 3-2 OPERATING SETUPS***

## 3.6 ADDRESS JUMPER GROUP

The Controller supports one of eight unique device addresses. When more than one Controller is used in a system, the address jumper on the Controller must be changed. Figure 3-3 shows the address jumper group located next to the terminator at position 5J; it also shows that terminal (pad) 0 is connected to terminal S. This is the factory-installed jumper, and it sets the Controller's address to 0.

In order to change this address, the factory-installed jumper (trace) must be cut. Then, a new jumper must be connected between terminal S and the selected address terminal. Figure 3-4 shows that the factory-installed jumper has been cut, and a new jumper has been installed between terminal S and address terminal 1. The address of the Controller is now 1.

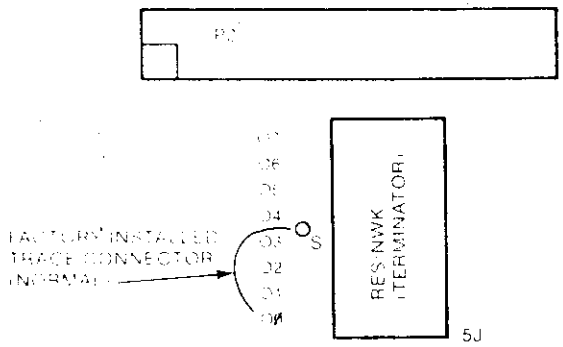



*FIGURE 3-3 NORMAL (FACTORY-INSTALLED) ADDRESS JUMPER*

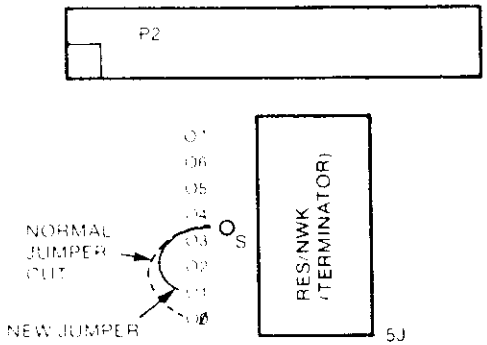



*FIGURE 3-4 CHANGED ADDRESS JUMPER (CONTROLLER 1)*



# CHAPTER 4
# THEORY OF OPERATION

## 4.1 GENERAL

This chapter discusses the theory of operation of the S1410 Controller and lays down the guidelines that will enable the user to use the Controller successfully in any number of applications.

### 4.1.1 Conventions

Signals or lines can be active in either a high or low state. The terms signal, signal lines, and lines mean the same thing. A low state is equivalent to a voltage level of 0.8 volts or less, and a high state is equivalent to a voltage level of 2.4 volts or more. Some texts use the term 'asserted' to mean active. In this manual, only the term active is used; if the term 'asserted' appears, it is only for reference.

### 4.1.2 Name and Abbreviations

The dash (-), or the lack of one, indicates the active state of a signal. The active state of a signal is that state which is required for a given operation. When a dash is appended to the end of a signal name, the signal is active when it is low. When no dash appears at the end of a signal name, the signal is active when it is high. Some signal lines have two so-called active (or significant) states. When the level on the line is high, a particular operation takes place. When the level on the line is low, a different operation takes place. The following examples show the use of these conventions.

BUSY- The signal BUSY- is active when it is at low level because it has the dash appended.

BUSY The signal BUSY is active at a high level because it does not have the dash appended.

C-/D The line C-/D (command-/data) has a dual purpose; the slash (/) indicates quality. The dash after the C indicates that when this line is at a low level, command mode is indicted and when it is at a high level, data mode is indicated.

Other designations used to define signal lines are listed below.

| | |
|---|---|
| Drv | Driver |
| Rcvr | Receiver |
| OC | Open collector |
| Tri-State | Line has three states: high, low, high impedance |
| 220/330 | Line termination: 220 Ohms to source voltage/330 Ohms to ground. |

### 4.1.3 Signal Definitions

The following tables list and define the signals that appear on the SASI Bus lines between the host adapter and the controller.

*TABLE 4-1 HOST BUS STATUS SIGNALS*

| NAME | DRV/RCVR | DEFINITION |
|---|---|---|
| I-/O | Drv OC | Input-/Output: The controller drives this line. A low level |

on this line indicates that the controller is driving the data in on the host bus. A high level on this line indicates that the host adapter is driving the data out on the host bus. The host adapter monitors this line and uses it to enable and disable its data bus drivers. This signal is qualified by signal REQ-.

| | | |
|---|---|---|
| C-/D | Drv OC | Command-/Data: This signal line indicates whether the information on the data bus consists of command or data bytes. A low means command bytes; a high means data bytes. This signal is qualified by signal REQ-. |
| BUSY- | Drv OC | Busy: The controller generates this active low signal in response to the SEL-signal and the address bit (DBO-to DB7-) from the host adapter. The busy signal informs the host adapter that the controller is ready to conduct transactions on the host bus. |
| MSG | Drv OC | Message The controller sends this active low signal to the host adapter to indicate that the current command has been completed When MSG-is active, the I-/O signal line is always low so that the controller can drive the bus data lines. This signal is qualified by signal REQ-. |

***TABLE 4-2 SUMMARY OF HOST BUS STATUS SIGNALS***

| I-/O | C-/D | MSG- | DEFINITION |
|---|---|---|---|
| High | Low | High | The controller receives command from the host adapter. |
| High | High | High | The controller receives data from the host adapter. |
| Low | High | High | The controller sends data to the host adapter. |
| Low | Low | High | The controller sends error status byte to the host adapter. |
| Low | Low | Low | The controller informs the host adapter that it has completed the current command. |

***TABLE 4-3 CONTROLLER - HOST HANDSHAKING***

| NAME | DRV/RCVR | DEFINITION |
|---|---|---|
| REQ- | Drv OC | Request: The controller sends this active low signal to the host adapter for each byte transferred across the interface This signal qualifies signals I-/O. C-/D and MSG-. |
| ACK- | Rcvr. 220/330 | Acknowledge The host adapter generates this active low signal in response to the REQ-signal from the controller when the host is ready to receive or transmit a byte of data In order to complete the handshake. the host adapter must send an acknowledge (ACK-) in response to each request (REQ-) from the controller. |

***TABLE 4-4 HOST BUS CONTROL SIGNALS***

| NAME | DRV/RCVR | DEFINITION |
|---|---|---|
| RST- | Rcvr. 220/220 | Reset. The host adapter sends this active low signal to the controller to force the controller to the idle state. After RST-has become active, any controller status is cleared. RST-also causes the deactivation of all signals to the drives. The time requirement for the RST-signal are as follows: |

| Minimum | Maximum |
|---|---|
| 100 nsec. | None |

| | | |
|---|---|---|
| SEL- | Rcvr, 220/330 | Select: The host adapter sends this active low signal to the controller to initiate a command transaction. Along with SEL-, the host adapter must also send an address bit to select the controller (DBO- for controller). The controller must not be busy. The host adapter must deactivate SEL-before the end of the current command. |

***TABLE 4-5 HOST BUS DATA SIGNALS***

| NAME | DRV/RCVR | DEFINITION |
|---|---|---|
| DB7- to | Tri-State. | These are the eight data bits (lines) of the host bus (DB0 = LSB). |
| DB0- | 220/330 | Each line is also used as address bits to select a controller in systems using multiple controllers (see Chapter 3). The normal connection (hardwired on the board) is to DB0-which is the address of controller 0. Any other connection requires cutting the existing trace on the board and adding a jumper. |

The following list show the bit assignments.

DB0- Controller 0
DB1- Controller 1
DB2- Controller 2
DB3- Controller 3
DB4- Controller 4
DB5- Controller 5
DB6- Controller 6
DB7- Controller 7

## 4.2 BASIC OPERATING CONFIGURATION

The basic operating configuration consists of a host adapter, S1410 Controller, and a Winchester disk drive with an interface that is compatible with that of the Seagate Technology ST506 disk drive. Figure 4-1 shows the basic setup. Also shown is an additional, optional drive; the controller can control a maximum of two drives.

The host can be one of a number of computer systems; the host adapter is an interface between the host's bus and the controller.

## 4.3 DETAILED DESCRIPTION (HANDSHAKING AND TIMING)

The following paragraphs describe the interaction between the controller and the host adapter.

### 4.3.1 Controller Selection

Before the host adapter can begin a transaction, it must select the controller. The host adapter selects the controller by activating the SEL-control signal and the address bit of the controller Any bit.

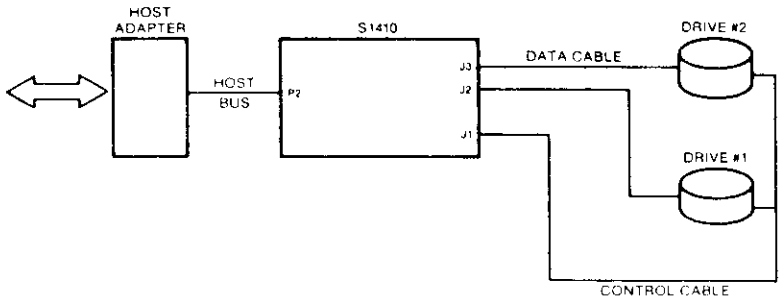



***FIGURE 4—1 BASIC OPERATING CONFIGURATON***

DB0-through DB7-, can be the address bit in a system with multiple controllers (all controllers leave the factory with DB0- connected to the controller's address logic). For this discussion, the controller's address is 0.

The timing diagram in Figure 4-2 shows the basic timing requirements. Upon receiving both the SEL- signal and DB0-, the controller activates the BUSY-signal. As shown in the timing diagram, both SEL-and DB0- must be active (low) before the controller can activate the BUSY- signal. During the selection process, the host has control of the data bus as signified by the deactivation of the I-/O line. Selection is complete when BUSY- becomes active. The SEL- signal must be deactivated by the host interface before the current controller operation has completed. It is recommended that the SEL- line be deactivated at or before the time the first command byte is sent to the controller. The controller then enters the command mode.

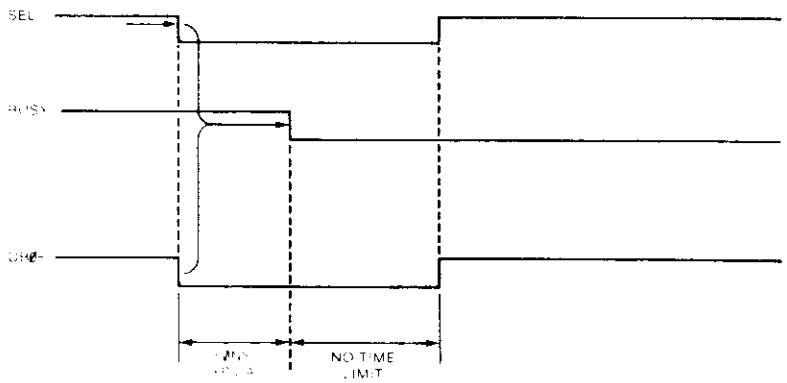



***FIGURE 4-2 CONTROLLER SELECT TIMING***

### 4.3.2 Command Mode

The controller receives commands from the host adapter using a handshaking sequence. The controller places a low level on the C-/D (command-/data) line to indicate that it wants a command from the host adapter and places a high level on the I-/O line to indicate that the movement of information is from the host adapter out to the controller. The MSG-line is high.

The controller activates the REQ- line within 10 microseconds after signals I-/O, C-/D and MSG- have been placed at high, low and high levels, respectively. The host adapter responds by activating the ACK- signal when a command byte is ready for the controller. The command byte placed on the data bus by the host must be stable within 250 nanoseconds after the ACK- signal is activated. The command byte must be held stable until REQ- is deactivated. The host deactivates ACK- after REQ- goes high. This completes the handshake for the first command byte. Each succeeding command byte from the host adapter requires the same complete handshake sequence. See Figure 4-4 for data bus, REQ-, and ACK- timing. See Table 4-2 for I-/O, C-/D and MSG- definition.

### 4.3.3 Data Transfer

The timing diagrams in Figures 4-3 and 4-4 illustrate the required timing for data transfer. See Table 4-2 for I-/O, C-/D and MSG-definition.

### 4.3.4 Status Bytes

Two bytes of status are passed to the host at the end of all commands. The first byte informs the host if any errors occurred during the execution of the command. The second is a zero byte. It signals to the host that the command is complete. Figure 4-3 shows the data bus, REQ- and ACK- timing. See Table 4-2 for I-/O, C-/D and MSG-definition. Figure 4-5a shows the format of these two bytes.

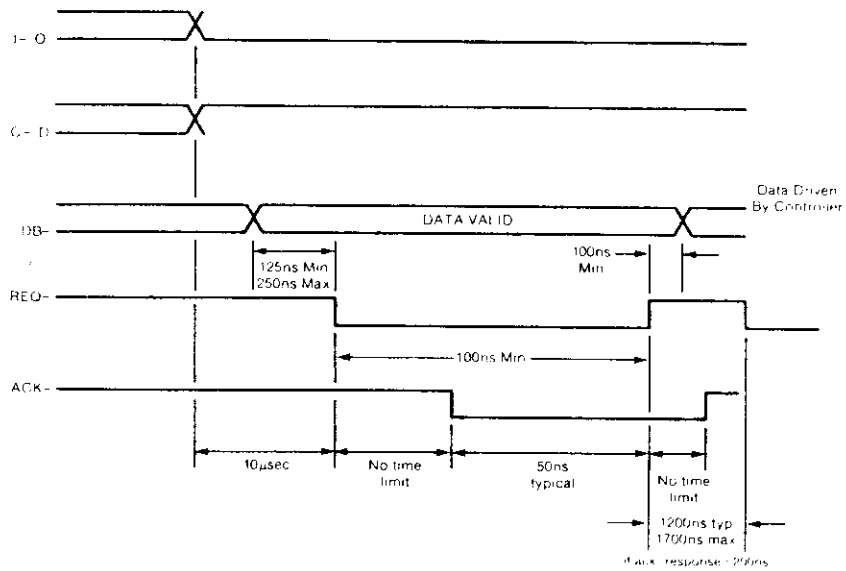



***FIGURE 4-3 DATA TRANSFER TO HOST, TIMING***

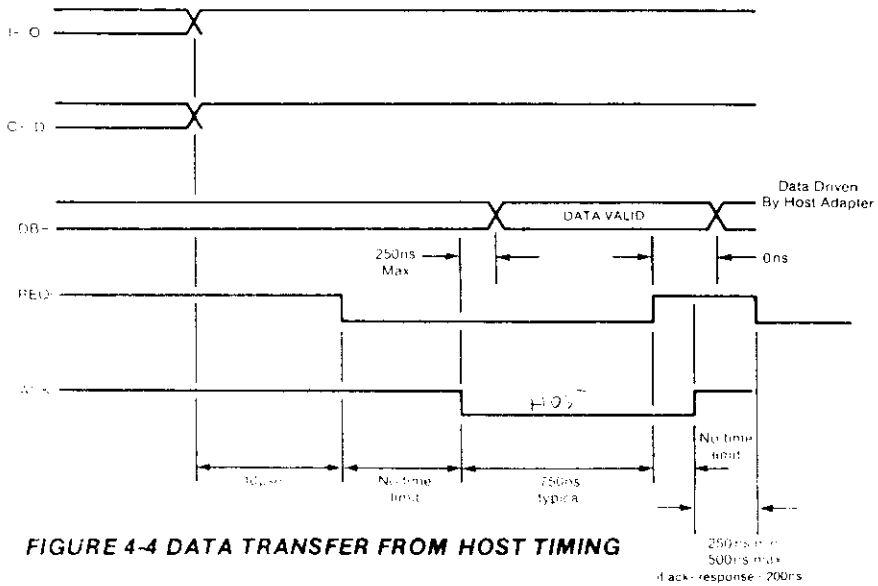



*FIGURE 4-4 DATA TRANSFER FROM HOST TIMING*

## 4.4 PROGRAMMING INFORMATION

The following section discusses communications between the controller and host from the point of view of the codes that are passed. The host sends commands to the controller through the host adapter. The controller then performs the commands and reports back to the host.

## 4.5 COMMANDS

The host sends a six-byte block to the controller to specify the operation. This block is the Device Control Block (DCB). Figure 4-5 shows the composition of the DCB. The list that follows Figure 4-5 defines the bytes that make up the DCB.

At the end of a command, the controller returns two completion status bytes to the host. The format of these bytes is shown in Figure 4-5a.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | Cmd class | | | Opcode | | | | |
| Byte 1 | LUN | | | High Address | | | | |
| Byte 2 | Middle Address | | | | | | | |
| Byte 3 | Low Address | | | | | | | |
| Byte 4 | Interleave or Block Count | | | | | | | |
| Byte 5 | Control Field | | | | | | | |

*FIGURE 4-5 DEVICE CONTROL BLOCK (DCB) FORMAT*

Byte 0 — Bits 7, 6 and 5 identify the class of the command. Bits 4 through 0 contain the opcode of the command.

Byte 1 — Bits 7, 6 and 5 identify the logical unit number (LUN). Bits 4 through 0 contain logical disk address 2.



Byte 2 — Bits 7 through 0 contain logical disk address 1.

Byte 3 — Bits 7 through 0 contain logical disk address 0 (LSB).

Byte 4 — Bits 7 through 0 specify the interleave or block count.

Byte 5 — Bits 7 through 0 contain the control field.

***Next to Last Status Byte***

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | d | 0 | 0 | 0 | ERR | 0 |

Bits 0, 2-4,6,7 — Set to zero.

Bit 1 — When set, error occurred during command execution.

Bit 5 — Logical unit number of drive, d-0 or 1.

***Last Status Byte***

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bits 0-7 — Set to zero

*FIGURE 4-5A COMPLETION STATUS BYTES*

### 4.5.1 Control Byte

The control field, byte 5, of the DCB allows the user to select options for several different types and makes of disk drives. The following list defines the bits of the control byte.

Bit 0 — Half-step option of Seagate and Texas Instrument drives.

Bit 1 — Half-step option for Tandon drives.

Bit 2 — Buffer-step option for drives made by Computer Memories, Inc. and Rotating Memories, Inc. (200 microsecond pulse per step).

Bit 3-5 — Spare. Set to zero for future use.

Bit 6 — If one, during a read sector command, the failing sector is not re-read on the next revolution before attempting correction. This bit should be set to zero for normal operation.

Bit 7 — Disable the four retries by the controller on all disk-access commands. Set this bit only during the evaluation of the performance of a disk drive. This bit should be set to zero for normal operation.

(#04)

#07: 15 usecs pulse step

**NOTE:** The step option bits (2-0) are mutually exclusive and only one option should be selected in any given configuration.

### 4.5.2 Logical Address (High, Middle and Low)

The logical address of the drive is computed by using the following equation.

Logical Address = (CYADR * HDCYL + HDADR) * SETRK + SEADR

Where:

CYADR = Cylinder Address
HDADR = Head Address
SEADR = Sector Address
HDCYL = Number of Heads per Cylinder
SETRK = Number of Sectors per Track

The commands fall into eight classes, 0 through 7; only classes 0 and 7 are used. Class 0 commands are data, non-data transfer, and status commands. Classes 1 through 6 are reserved. Class 7 are diagnostic commands.

Each command is described below. The description includes its class, opcode, and format. When a slash (/) represents a bit position, the slash means that the value of that bit is not important (a don't-care bit).

### 4.5.3 Test Drive Ready (Class 0, Opcode 00)

This command selects a particular drive and verifies that the drive is ready. The following diagram shows the format of the device control block for this command.

d = drive, 0 or 1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Byte 1 | 0 | 0 | d | / | / | / | / | / |
| Byte 2 | / | / | / | / | / | / | / | / |
| Byte 3 | / | / | / | / | / | / | / | / |
| Byte 4 | / | / | / | / | / | / | / | / |
| Byte 5 | / | / | / | / | / | / | / | / |

### 4.5.4 Recalibrate (Class 0, Opcode 01)

This command positions the read/write (R/W) arm to track 00.

d = drive, 0 or 1
r = retries
s = step option

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Byte 1 | 0 | 0 | d | / | / | / | / | / |
| Byte 2 | / | / | / | / | / | / | / | / |
| Byte 3 | / | / | / | / | / | / | / | / |
| Byte 4 | / | / | / | / | / | / | / | / |
| Byte 5 | r | 0 | 0 | 0 | 0 | s | s | s |



### 4.5.5 Reserved (Class 0, Opcode 02)

This opcode is not used.

### 4.5.6 Request Sense Status (Class 0, Opcode 03)

The host must send this command immediately after it detects an error. The command causes the Controller to return four bytes of drive and Controller status; the formats of these four bytes are shown after the DCB. When an error occurs on a multiple sector data transfer, (read or write), the Request Sense Status command returns the logical address of the failing sector in bytes 1, 2 and 3. If the Request Sense Status command is issued after any of the Format commands or the Check Track Format command, then the logical address returned by the controller points to one sector beyond the last track formatted or checked if there was no error. If there was an error, then the logical address returned points to the track in error. The tables that follow the formats list the error codes.

d = drive, 0 or 1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| Byte 1 | 0 | 0 | d | / | / | / | / | / |
| Byte 2 | / | / | / | / | / | / | / | / |
| Byte 3 | / | / | / | / | / | / | / | / |
| Byte 4 | / | / | / | / | / | / | / | / |
| Byte 5 | / | / | / | / | / | / | / | / |

***SENSE BYTES***

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | SEE BELOW | | | | | | | |

| | |
|---|---|
| Bits 0-3 | Error Code |
| Bits 4 5 | Error Type |
| Bit 6 | Spare, set to zero |
| Bit 7 | Address valid, when set |

The address valid bit in the error code byte (bit 7) is relevant only when the previous command required a logical block address in which case it is always returned as a one otherwise it is set to zero. For instance, if a Recalibrate command is followed immediately by a Request Sense Status command, the address valid bit would be returned as zero since this command does not require a logical block address to be passed in its DCB.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 1 | 0 | 0 | d | High Address | | | | |
| Byte 2 | Middle Address | | | | | | | |
| Byte 3 | Low Address | | | | | | | |

d = drive, 0 or 1

### TABLE 4-6 TYPE 0 ERROR CODES, DISK DRIVE

| HEX CODE | DEFINITION |
|---|---|
| 0 | The controller detected no error during the execution of the previous operation. |
| 1 | The controller did not detect an index signal from the drive. |
| 2 | The controller did not get a seek complete signal from the drive after seek operation. |
| 3 | The controller detected a write fault from drive during last operation. |
| 4 | After the controller selected the drive, the drive did not respond with ready signal. |
| 5 | Not used. |
| 6 | After stepping maximum number of cylinders, controller did not receive track 00 signal from the drive. |

### TABLE 4-7 TYPE 1 ERROR CODES, CONTROLLER

| HEX CODE | DEFINITION |
|---|---|
| 0 | ID Read Error: The controller detected an ECC error in the target ID field on the disk. |
| 1 | Data Error: The controller detected an uncorrectable ECC error in the target sector during a read operation. |
| 2 | Address Mark: The controller did not detect the target address mark (AM) on the disk. |
| 3 | Not used. |
| 4 | Sector Not Found: The controller found the correct cylinder and head, but not the target sector. |
| 5 | Seek Error: The controller detected an incorrect cylinder or track, or both. |
| 6 | Not used. |
| 7 | Not used. |
| 8 | Correctable Data Error: The controller detected a correctable ECC error in the target data field. |
| 9 | Bad Track: The controller detected the bad track flag during the last operation. |
| A | Format Error: During a check-track command, the controller detected one of the following:<br>1) track not formatted<br>2) wrong interleave<br>3) ID ECC error on at least one (1) sector |



### TABLE 4-8
### TYPES 2 AND 3 ERROR CODES, COMMAND AND MISCELLANEOUS

| HEX CODE | TYPE | DEFINITION |
|---|---|---|
| 0 | 2 | Invalid Command: The controller has received an invalid command from the host. |
| 1 | 2 | Illegal Disk Address: The controller detected an address that is beyond the maximum range. |
| 0 | 3 | RAM Error: The controller detected a data error during the RAM sector buffer diagnostic. |
| 1 | 3 | Program Memory Checksum Error: During its internal diagnostic, the controller detected a program-memory checksum error. |
| 2 | 3 | ECC Polynominal Error: During the controller's internal diagnostic, the hardware ECC generator failed its test. |

The following is a summary of the error codes returned as the result of the Request Sense Status command.

**NOTE:** The address valid bit (bit 7) may or may not be set and is not included here for clarity.

| Error Code (hex) | Meaning. |
|---|---|
| 00 | No error detected (command completed ok). |
| 01 | No index detected from disk drive. |
| 02 | No seek complete from disk drive. |
| 03 | Write fault from disk drive. |
| 04 | Drive not ready after it was selected. |
| 05 | Not used. |
| 06 | Track 00 not found. |
| 07-0F | Not used. |
| 10 | ID field read error |
| 11 | Uncorrectable data error. |
| 12 | Address mark not found. |
| 13 | Not used. |
| 14 | Target sector not found. |
| 15 | Seek error. |
| 16-17 | Not used. |
| 18 | Correctable data error. |
| 19 | Bad track flag detected. |
| 1A | Format error. |
| 1B-1F | Not used. |
| 20 | Invalid command. |
| 21 | Illegal disk address. |
| 22-2F | Not used. |
| 30 | Ram diagnostic failure. |
| 31 | Program memory checksum error. |
| 32 | ECC diagnostic failure. |
| 33-3F | Not used. |

#### 4.5.7 Format Drive (Class 0, Opcode 04)

This command formats all sectors with ID and data fields according

to the selected interleave factor. Also, it writes 6C Hex into data fields. The starting address is passed in the DCB. The Controller will format from the starting address to the end of the disk.

d = drive, 0 or 1

r = retries

s = step option

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Byte 1 | 0 | 0 | d | High Address | | | | |
| Byte 2 | Middle Address | | | | | | | |
| Byte 3 | Low Address | | | | | | | |
| Byte 4 | 0 | 0 | 0 | Interleave | | | | |
| Byte 5 | r | 0 | 0 | 0 | 0 | s | s | s |

Interleave:
1 to 31 for 256 sectors
1 to 16 for 512 sectors } Refer to Section 4.10

### 4.5.8 Check Track Format (Class 0, Opcode 05)

This command checks the format on the specified track for correct ID and interleave. The command does not read the data field.

d = drive, 0 or 1

r = retries

s = step option

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| Byte 1 | 0 | 0 | d | High Address | | | | |
| Byte 2 | Middle Address | | | | | | | |
| Byte 3 | Low Address | | | | | | | |
| Byte 4 | 0 | 0 | 0 | Interleave | | | | |
| Byte 5 | r | 0 | 0 | 0 | 0 | s | s | s |

Interleave:
1 to 31 for 256 sectors
1 to 16 for 512 sectors } Refer to Section 4.10

### 4.5.9 Format Track (Class 0, Opcode 06)

This command formats a specified track and can be used to clear bad sector flag in all sectors on the specified track that was



previously formatted with the Format Bad Track command. The command writes 6C Hex into all data fields.

d = drive, 0 or 1

r = retries

s = step option

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| Byte 1 | 0 | 0 | d | High Address | | | | |
| Byte 2 | Middle Address | | | | | | | |
| Byte 3 | Low Address | | | | | | | |
| Byte 4 | 0 | 0 | 0 | Interleave | | | | |
| Byte 5 | r | 0 | 0 | 0 | 0 | s | s | s |

Interleave:
1 to 31 for 256 sectors
1 to 16 for 512 sectors } Refer to Section 4.10

### 4.5.10 Format Bad Track (Class 0, Opcode 07)

This command formats the specified track and sets the bad-sector flag in the ID fields. It does not write the data fields.

d = drive, 0 or 1

r = retries

s = step option

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| Byte 1 | 0 | 0 | d | High Address | | | | |
| Byte 2 | Middle Address | | | | | | | |
| Byte 3 | Low Address | | | | | | | |
| Byte 4 | 0 | 0 | 0 | Interleave | | | | |
| Byte 5 | r | 0 | 0 | 0 | 0 | s | s | s |

Interleave:
1 to 31 for 256 sectors
1 to 16 for 512 sectors } Refer to Section 4.10

### 4.5.11 Read (Class 0, Opcode 08)

This command reads the specified number of sectors, starting with the initial sector address contained in the DCB.

d = drive, 0 or 1

r = retries

a = retry option on data ECC error

s = step option

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Byte 1 | 0 | 0 | d | High Address | | | | |
| Byte 2 | Middle Address | | | | | | | |
| Byte 3 | Low Address | | | | | | | |
| Byte 4 | Block Count | | | | | | | |
| Byte 5 | r | a | 0 | 0 | 0 | s | s | s |

### 4.5.12 Reserved (Class 0 Opcode 09)

This opcode is not used.

### 4.5.13 Write (Class 0, Opcode 0A)

This command writes the specified number of sectors, starting with the initial sector address contained in the DCB. Each sector of data can be either 256 or 512 bytes long. The sector size is jumper selectable, see Chapter 3.

d = drive, 0 or 1

r = retries

s = step option

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| Byte 1 | 0 | 0 | d | High Address | | | | |
| Byte 2 | Middle Address | | | | | | | |
| Byte 3 | Low Address | | | | | | | |
| Byte 4 | Block Count | | | | | | | |
| Byte 5 | r | 0 | 0 | 0 | 0 | s | s | s |

### 4.5.14 Seek (Class 0, Opcode 0B)

This command initiates a seek to the track specified in the DCB. The drive must be formatted.

d = drive, 0 or 1

r = retries

s = step option

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| Byte 1 | 0 | 0 | d | High Address | | | | |
| Byte 2 | Middle Address | | | | | | | |
| Byte 3 | Low Address | | | | | | | |
| Byte 4 | / | / | / | / | / | / | / | / |
| Byte 5 | r | 0 | 0 | 0 | 0 | s | s | s |



### 4.5.15 Initialize Drive Characteristics (Class 0, Opcode 0C)

This command enables the user to configure the Controller to work with drives that have different capacities and characteristics. However, both drive 0 and drive 1 must be of the same manufacturer and model number.

After the host sends the command (DCB) to the controller, it then sends an eight-byte block of data that contains the drive parameters. Some of the parameters occupy two bytes; all two-byte parameters are transferred with the most significant byte (MSB) first. The eight bytes are listed below.

C= Maximum number of cylinders (2 bytes)
H= Maximum number of heads (1 byte)
W= Starting reduced write current cylinder (2 bytes)
P= Starting write precompensation cylinder (2 bytes)
E= Maximum ECC data burst length (1 byte)

When the controller is powered up or reset, the default values listed below are set.

Maximum number of cylinders = ~~153~~ → 320
Maximum number of heads = ~~4~~ 8
Starting reduced write current cylinder = ~~128~~ 132
Starting write precompensation cylinder = ~~64~~ 0
Maximum ECC data burst length = 11 bits

The configuration parameters for various disk drives can be found at the begining of Appendix A in the back of this manual.

The parameter for the maximum ECC burst length defines the length of a burst error in the data field that the controller will correct. The burst length is defined as the number of bits from first error bit to the last error bit. For example, the controller detected a 5-bit ECC error and the erroneous data appeared as C5 (1100 0101) before correction and could appear as D4 (1101 0100) after the correction. However, if the host has set the maximum ECC burst length at 4 bits, the controller would have to flag this data as uncorrectable. This is a type 1, code 1 error.

Command Bytes

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| Byte 1 | / | / | / | / | / | / | / | / |
| Byte 2 | / | / | / | / | / | / | / | / |
| Byte 3 | / | / | / | / | / | / | / | / |
| Byte 4 | / | / | / | / | / | / | / | / |
| Byte 5 | / | / | / | / | / | / | / | / |

Parameter Bytes

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | C | C | C | C | C | C | C | C |
| Byte 1 | C | C | C | C | C | C | C | C |
| Byte 2 | 0 | 0 | 0 | 0 | H | H | H | H |
| Byte 3 | W | W | W | W | W | W | W | W |
| Byte 4 | W | W | W | W | W | W | W | W |
| Byte 5 | P | P | P | P | P | P | P | P |
| Byte 6 | P | P | P | P | P | P | P | P |
| Byte 7 | 0 | 0 | 0 | 0 | E | E | E | E |

### 4.5.16 Read ECC Burst Error Length (Class 0, Opcode 0D)

This command transfers one byte to the host. This byte contains the value of the ECC burst length that the controller detected during the last Read command. This byte is valid only after a correctable ECC data error, type 1, code 8.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| Byte 1 | | | | | | | | |
| Byte 2 | | | | | | | | |
| Byte 3 | | | | | | | | |
| Byte 4 | | | | | | | | |
| Byte 5 | | | | | | | | |

### 4.5.17 RAM Diagnostic (Class 7, Opcode 00)

This command performs a data pattern test on the RAM buffer.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| Byte 1 | | | | | | | | |
| Byte 2 | | | | | | | | |
| Byte 3 | | | | | | | | |
| Byte 4 | | | | | | | | |
| Byte 5 | | | | | | | | |

### 4.5.18 Reserved (Class 7, Opcode 01)

This opcode is not used

### 4.5.19 Reserved (Class 7, Opcode 02)

This opcode is not used



### 4.5.20 Drive Diagnostic (Class 7, Opcode 03)

This command tests both the drive and the drive-to-controller interface. The controller sends recalibrate and seek commands to the selected drive and verifies sector 0 of all the tracks on the disk. The controller does not perform any write operations during this command; it is assumed that the disk has been previously formatted.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| Byte 1 | 0 | 0 | d | | | | | |
| Byte 2 | | | | | | | | |
| Byte 3 | | | | | | | | |
| Byte 4 | | | | | | | | |
| Byte 5 | r | 0 | 0 | 0 | 0 | s | s | s |

### 4.5.21 Controller Internal Diagnostics (Class 7, Opcode 04)

This command causes the controller to perform a self-test. The controller checks its internal processor, data buffers, ECC circuitry, and the checksum of the program memory. The controller does not access the disk drive.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| Byte 1 | | | | | | | | |
| Byte 2 | | | | | | | | |
| Byte 3 | | | | | | | | |
| Byte 4 | | | | | | | | |
| Byte 5 | | | | | | | | |

### 4.5.22 Read Long (Class 7, Opcode 05)

This command transfers the target sector and four bytes of data ECC to the host. If an ECC error occurs during the read, the controller does not attempt to correct the data field. This command is useful in recovering data from a sector that contains an uncorrectable ECC error. It is also useful during diagnostic operations.

d = drive, 0 or 1

r = retries

s = step option

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| Byte 1 | 0 | 0 | d | High Address | | | | |
| Byte 2 | Middle Address | | | | | | | |
| Byte 3 | Low Address | | | | | | | |
| Byte 4 | Block Count | | | | | | | |
| Byte 5 | r | 0 | 0 | 0 | 0 | s | s | s |

### 4.5.23 Write Long (Class 7, Opcode 06)

This command transfers a sector of data and four appended ECC bytes to the disk drive. During this write operation, the host supplies the four ECC bytes instead of the usual hardware-generated ECC bytes. This command is useful only for diagnostic operations.

d = drive, 0 or 1

r = retries

s = step option

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| Byte 1 | 0 | 0 | d | High Address | | | | |
| Byte 2 | Middle Address | | | | | | | |
| Byte 3 | Low Address | | | | | | | |
| Byte 4 | Block Count | | | | | | | |
| Byte 5 | r | 0 | 0 | 0 | 0 | s | s | s |

## 4.6 SECTOR FORMAT

Figure 4-6 shows the format of the sector and the names of the fields of the information traveling over the Controller-drive interface. Table 4-9 lists the fields and a description of each field.

***TABLE 4-9 SECTOR FIELD DESCRIPTION***

| FIELD | BYTES | FIELD DESCRIPTION |
|---|---|---|
| AM | 4 | Address Mark |
| GAP1 | 9 | Zero Byte Gap |
| SYNC1 | 1 | ID Sync Byte |
| GAP2 | 2 | ID Zero Byte Gap |
| COM | 1 | ID Compare Byte |
| CYLH | 1 | Cylinder High (MSB) |
| CYLL | 1 | Cylinder Low (LSB) |
| HEAD | 1 | Head Number |
| SEC | 1 | Sector Number |
| FLAG | 1 | Flag Byte |
| ZER | 1 | Zero Byte |
| ECC1 | 4 | ID ECC Bytes |



***TABLE 4-9 SECTOR FIELD DESCRIPTION*** (con't)

| FIELD | BYTES | FIELD DESCRIPTION |
|---|---|---|
| GAP3 | 16 | Zero Byte Gap |
| SYNC2 | 1 | Data Field Sync Byte |
| GAP4 | 2 | Data Field Zero Byte Gap |
| DATA | 256/512 | Data Field |
| ECC2 | 4 | Data Field Ecc Bytes |
| GAP5 | 14/43 | Inter-record Zero Gap |

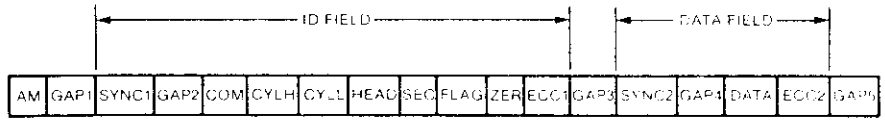



***FIGURE 4-6 SECTOR FORMAT***

## 4.7 EXECUTION OF DIAGNOSTICS

Since all of the diagnostics are not executed by the controller on power up, it is suggested that they be envoked by the host in the following order:

1) Controller internal diagnostics (Command code E4). This diagnostic tests all the logical and decision making capability of the controller as well as the program memory checksum and the error detection and correction circuits (ECC). Execution of this diagnostic ensures that the controller can communicate with the host.

2) Ram Diagnostic (Command code E0) should be executed next. This command verifies that the sector buffer is operational by writing, reading and verifying various data patterns to and from all locations.

3) If the parameters of the connected drives are different than the default parameters, see Section 4.5.15, the new configuration must be sent to the controller using the Initialize Drive Characteristics command (Command code 0C) before the Drive Diagnostic is executed. (See appendix A for the configuration parameters of various disk drives).

4) Before the Drive Diagnostic is executed, the host program should continuously issue a Test Drive Ready command to the controller (Command code 00) with the appropriate time-out until the drive becomes ready.

5) Drive Diagnostic (Command code E3). This diagnostic issues a Recalibrate to the disk drive and then steps through all tracks verifying the ECC on the identifier fields of the first sector of each track. If this diagnostic passes, it implies that the disk has been formatted and that the first ID field of each track is good.

## 4.8 ERROR CORRECTION PHILOSOPHY

Since the typical error correction time of the S1410 controller is approximately 50 milliseconds and therefore greater than the time for one revolution of the disk, the sector in error is optionally re-read (if bit 6 is not set in byte 5 of the read command DCB) on the next revolution during a Read command. In most

cases, the error will be soft and will not reappear on the re-read. This initial re-read of the failing sector is over and above the retry count passed in the DCB (bit 7, byte 5).

The retry count on errors is preset to 4 by the controller each time a sector has been read successfully. On a multiple sector transfer if an uncorrectable error was detected but subsequently found to be correctable on a retry, the retry count is reset to 4 before the next sector is read from the disk.

## 4.9 ALTERNATE TRACK ASSIGNMENT AND HANDLING

The assignment of alternate tracks and the lockout of bad tracks has to be done by the host computer. Bad areas on the disk are labeled defective on a track basis by issuing a Format Bad Track command (Command code 07). One procedure for assignment and handling of alternate tracks might be as follows:

1) The entire disk drive is formatted by issuing a Format Disk command (Command code 04) starting at logical track zero.

   If an error occurs during formatting then a Request Sense Status command should be issued. If a format error is indicated, bytes 1, 2 and 3 of the returned status gives the address of the bad track and a Format Bad Track command (command code 07) should be issued to that track. A new Format command is then issued to format the rest of the disk starting at one track beyond the bad track. If any other errors occur during the subsequent formatting, the above process should be repeated until the entire disk is formatted.

2) A recalibrate command is issued (Command code 01) to position the heads over track 00.

3) All sectors on the disk are read to see if any uncorrectable ECC errors have occured in the data. The Format command places a 6C (hex) pattern in the data fields of all sectors and the host program can optionally verify this data pattern after the data is read into memory, however, verifying the data byte for byte is not normally necessary since the Error Detection and Correction circuitry flags all uncorrectable errors. If a large block of host memory is available, multiple sector reads can be issued to speed up the verify process.

4) When an uncorrectable error is found, a Format Bad Track command (Command code 07) is issued to the failing track which writes a bad track flag into all identifier fields. Whenever this track is subsequently accessed, an error will result and the Sense Status command which follows will indicate an error code 19 (hex). It should be noted that once a track has been flagged as defective, the data fields cannot be accessed.

   The host program must maintain a table of bad tracks and the alternates assigned to them. This table can be placed on a known good area of the disk (most manufacturers guarantee certain tracks to be error free).

   Whenever a user program accesses the disk, it should not be allowed by the operating system to issue a read or write command to the alternate tracks. The disk controller has no way of knowing when an alternate track is being read.



The alternate tracks are sometimes assigned at the end of the disk (highest track numbers) but they may be assigned to any tracks as long as the track table is maintained by the host.

In general, if four tracks are reserved as alternates, they should be adequate for all disk drives currently available given the error correction capability of the S1410 controller, however, it is recommended that the system programmer consult the disk drive manual for the hard defect specifications.

## 4.10 SECTOR INTERLEAVING

Variable sector interleaving is supported by the S1410 disk controller. When any format command is issued, any interleave value up to the number of sectors-per-track minus one may be passed in the Device Control Block (DCB byte 4). The interleave factor may be adjusted for maximum system performance. Interleaving allows logical contiguous sectors of data on a given track to be mapped onto non-adjacent physical sectors. An interleave factor of five, for instance means that every fifth physical sector is transferred as the next contiguous logical sector of data. It does not mean that five sectors of data are transferred in one revolution. If the operation is read and the interleave factor is five then a sector of data is read into the sector buffer first and during the time that the heads are passing over the next four physical sectors of the disk, the data is being transferred to the host. If the host cannot transfer the full sector of data during the four sector times available, then the controller has to wait a full revolution before the next logical sector can be read from the disk. If this happens, the interleave factor is too low and should be increased until an increase in operating system speed is noticed.

In order to take full advantage of the interleaving feature of the controller, the operating system should perform multiple sector data transfers. If single sector transfers are employed, the difference in speed with various interleave factors may not be dramatic.

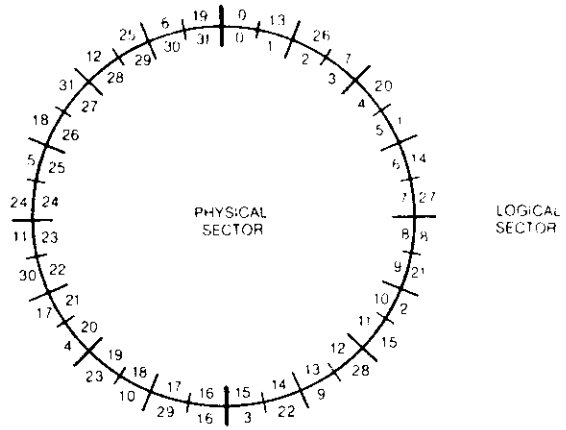



***FIGURE 4-10 TRACK FORMAT EXAMPLE OF 32 SECTORS-PER-TRACK WITH AN INTERLEAVE FACTOR OF 5***

# APPENDIX A

## CONFIGURATION PARAMETERS FOR VARIOUS DISK DRIVES

The following is a list of the configuration parmeters for various disk drives which are supported by the S1410 disk controller. The table lists the actual values to be pased to the controller in the Initialize Drive Characteristics command (0C). The first value of each entry is in decimal and the value following it in parentheses is its equivalent in hexidecimal. Note that the number of cylinders and heads passed is an absolute value (1 relative) and the reduce write current and write precomposition cylinder values are zero relative.

The abbreviations for the manufacturers are as follows:

| | |
|---|---|
| CMI | Computer Memories Incorporated. |
| OLI | Olivetti. |
| RMS | Rotating Memory Systems Incorporated. |
| SEA | Seagate Technology Incorporated. |
| TAN | Tandon Incorporated. |
| TI | Texas Instruments |
| RO | Rodime Ltd. |
| MS | Miniscribe |

| Mfgr/Model No. | Cylinders | Heads | Reduce Write Cyl. | Write Precomp Cyl. |
|---|---|---|---|---|
| CMI CM-5205 | 256 (100) | 2 | 256 (100) | 256 (100) |
| CMI CM-5410 | 256 (100) | 4 | 256 (100) | 256 (100) |
| CMI CM-5616 | 256 (100) | 6 | 256 (100) | 256 (100) |
| OLI HD561 | 180 (B4) | 2 | 128 (80) | 180 (B4) |
| OLI HD562 | 180 (B4) | 2 | 128 (80) | 180 (B4) |
| RMS 503 | 153 (99) | 2 | 77 (4B) | 77 (4B) |
| RMS 506 | 153 (99) | 4 | 77 (4B) | 77 (4B) |
| RMS 512 | 153 (99) | 8 | 77 (4B) | 77 (4B) |
| SEA ST-506 | 153 (99) | 4 | 128 (80) | 64 (40) |
| SEA ST-412 | 306 (132) | 4 | 128 (80) | 64 (40) |
| TAN TM602S | 153 (99) | 4 | 128 (80) | 153 (99) |
| TAN TM603S | 153 (99) | 6 | 128 (80) | 153 (99) |
| TAN TM603SE | 230 (E6) | 6 | 128 (80) | 128 (80) |
| TI 5¼+ | 153 (99) | 4 | 64 (40) | 64 (40) |
| RO 101 | 192 (C0) | 2 | 96 (60) | 0 (0) |
| RO 102 | 192 (C0) | 4 | 96 (60) | 0 (0) |
| RO 103 | 192 (C0) | 6 | 96 (60) | 0 (0) |
| RO 104 | 192 (C0) | 8 | 96 (60) | 0 (0) |
| RO 201 | 321 (141) | 2 | 132 (84) | 0 (0) |
| RO 202 | 321 (141) | 4 | 132 (84) | 0 (0) |
| RO 203 | 321 (141) | 6 | 132 (84) | 0 (0) |
| RO 204 | 321 (141) | 8 | 132 (84) | 0 (0) |
| MS 1-006 | 306 (132) | 2 | 153 (99) | 0 (0) |
| MS 1-012 | 306 (132) | 4 | 153 (99) | 0 (0) |

# APPENDIX B

## SAMPLE HARDWARE INTERFACE AND PROGRAMMING EXAMPLES FOR A Z-80 BASED MICROPROCESSOR TO INTERFACE TO THE XEBEC S1410 CONTROLLER

| SASI read port | Function. |
|---|---|
| 0 | Read data (from controller to CPU) |
| 1 | Read hardware status lines (from cntlr to CPU) |
| 2 | Not used |
| 3 | Not used |

| SASI write port | Function. |
|---|---|
| 0 | Write data (from CPU to controller) |
| 1 | Controller reset (and interface reset) |
| 2 | Generate controller select pulse |
| 3 | Not used |

### WRITE TRANSPARENT LATCH (DATA FROM HOST).

The data to be sent to the controller is passed through this register when the enable line is high and is latched up when the enable goes low. The latch enable is generated by a programmed I/O write to SASI write port 0 from the host. The positive going (trailing) edge of the negative true I/O write signal from the host also sets the handshake acknowledge flip flop which asserts the acknowledge signal (ACK-) on the SASI bus indicating to the controller that data is available from the host.

### WRITE DATA BUS DRIVER (TO THE CONTROLLER).

This is a tri-state bus driver which is enabled to drive the SASI bus under command of the controller when it brings the Input-/Output signal on the SASI bus high.

### READ DATA BUS RECEIVER (FROM THE CONTROLLER).

Data or Status from the controller is driven onto the host data bus by a programmed I/O read (SASI read port 0) from the host.

### SASI HARDWARE STATUS LINE RECEIVER (FROM CONTROLLER).

The hardware status is driven onto the host data bus when an I/O read is issued by the host program to SASI read port 1. It should be noted that the status signals from the SASI bus are inverted before they are sent to the host.

The bits in the hardware status as seen by the host program are defined as follows:*

| Bit | Usage |
|---|---|
| 7 | Not used (MSB). |
| 6 | Not used. |
| 5 | Not used. |
| 4 | Input/Output- signal from the contoller.<br>1 = input from controller to host, 0 = output from host to controller |

3 Command/Data- signal from the controller.
1 = Command info on SASI bus, 0 = data on SASI bus.

2 Message (MSG-) signal from the controller.
1 = second completion status byte.

1 BUSY- signal from the controller.
1 = controller busy.

0 Request signal from the controller (REQ-).
1 = transfer request.

***NOTE:** The polarities of these signals do not match those of the S1410 because in this sample circuit they are inverted by the SASI status interface chip.

## HANDSHAKE ACKNOWLEDGE FLIP FLOP.

This flip flop is used for all data, command or status handshakes between the host and the controller under programmed I/O control. As long as the controller is not selected, this flip flop is kept directly cleared by the inactive REQ-line which in turn keeps the acknowledge line (ACK-) inactive to the controller.

After the controller is selected and ready to accept a command from the host, it asserts the REQ- line signaling the host to write a command byte into the transparent latch. When the host writes into the latch, the handshake acknowledge flip flop is set signaling the controller that the data is available. When the controller has read the command byte from the SASI bus, it deactivates the REQ-line clearing the handshake acknowlege flip flop, which in turn deactivates the acknowledge line (ACK-).

When the ACK- signal is deasserted, the REQ-signal is again activated by the controller starting the handshake sequence for the next byte.

## PROGRAMMING EXAMPLES.

The S1410 controller commands fall into two classes:

A Class 0
1) — Non data transfer commands

2) — Data transfer commands

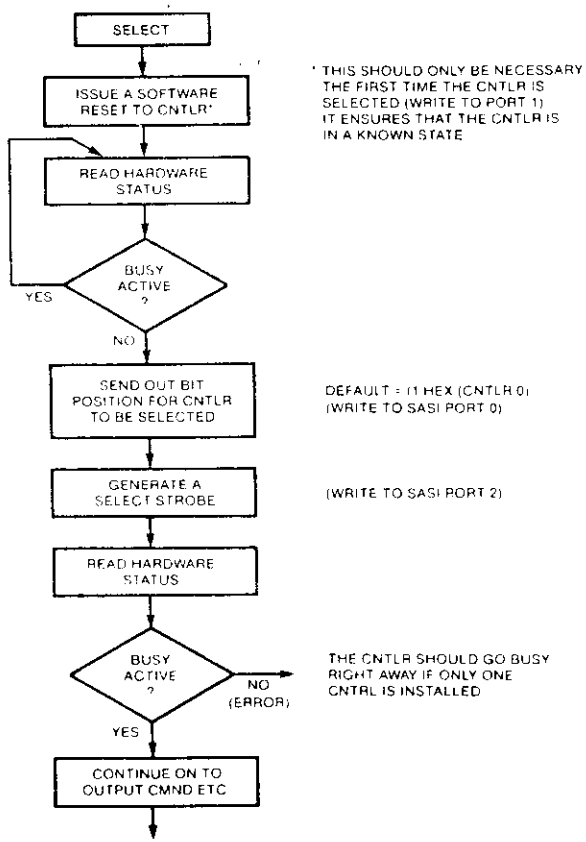
3) — Status command

B Class 7
1) Diagnostics

Before a command is sent, the controller has to be selected.



## CONTROLLER SELECTION.




## NON DATA TRANSFER COMMANDS.

The Test for Drive Ready command is shown here as an example of a non data transfer command. Some of the other commands which fall into this category are:

1) — Recalibrate

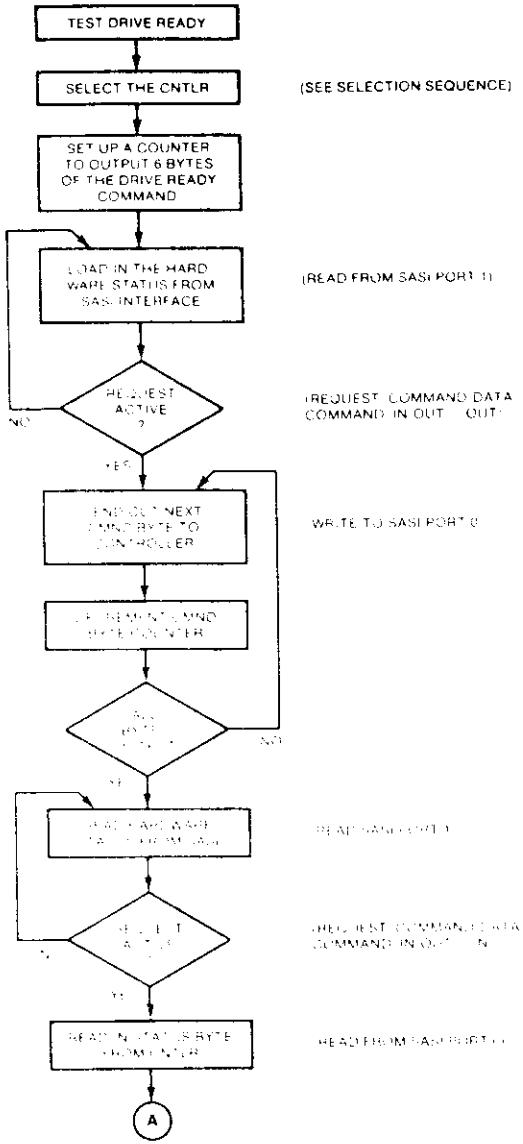
2) — Format Disk Drive

3) -- Check Track Format

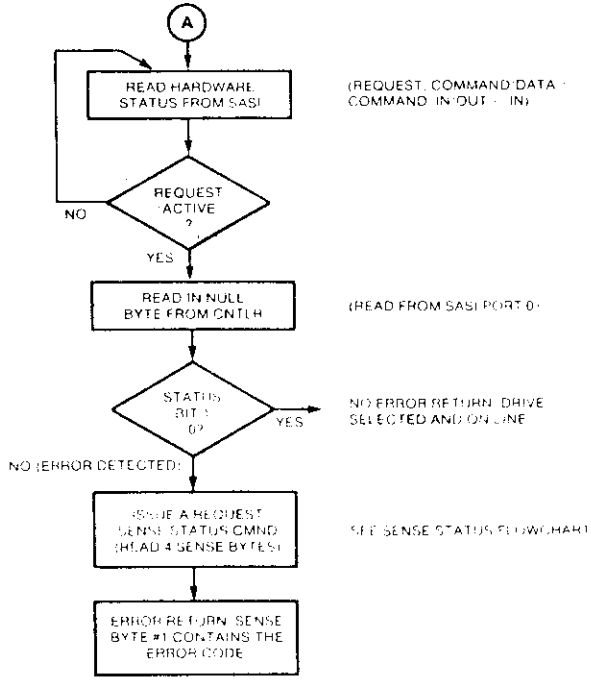
4) -- Seek

5) -- Various diagnostics

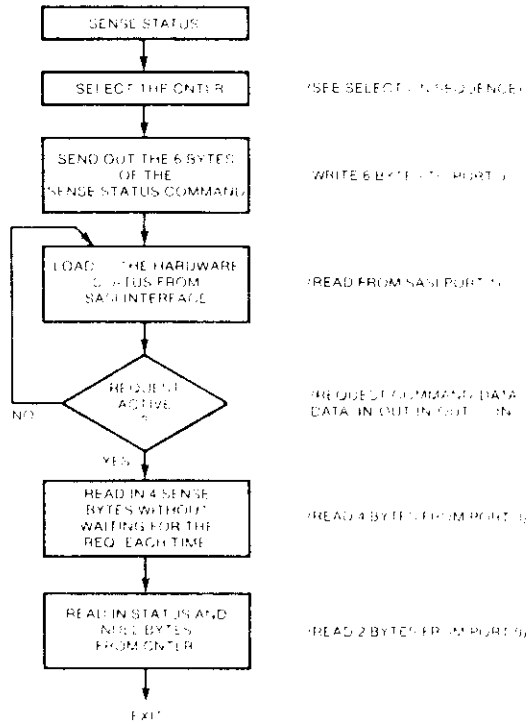
This command selects a specified disk drive and verifies that it is ready. This particular command should be issued initially to make sure all disk drives to be accessed during system operation are operational.



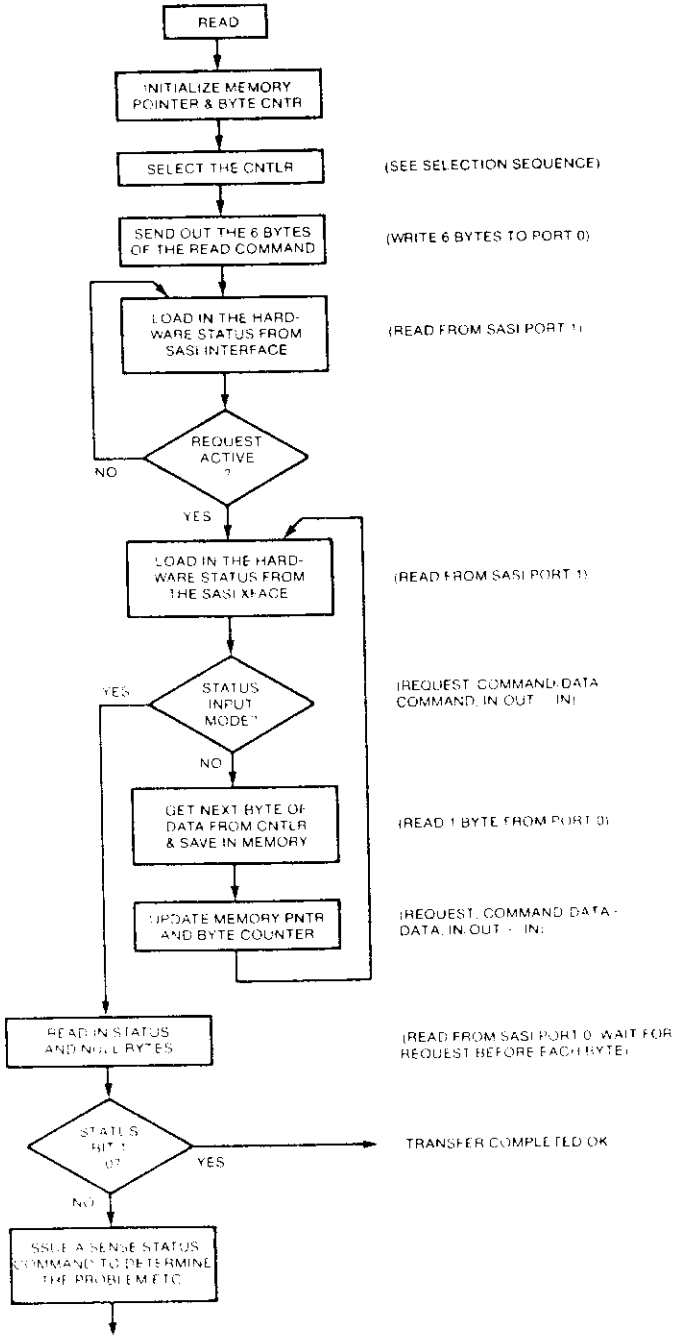



## SENSE STATUS COMMAND

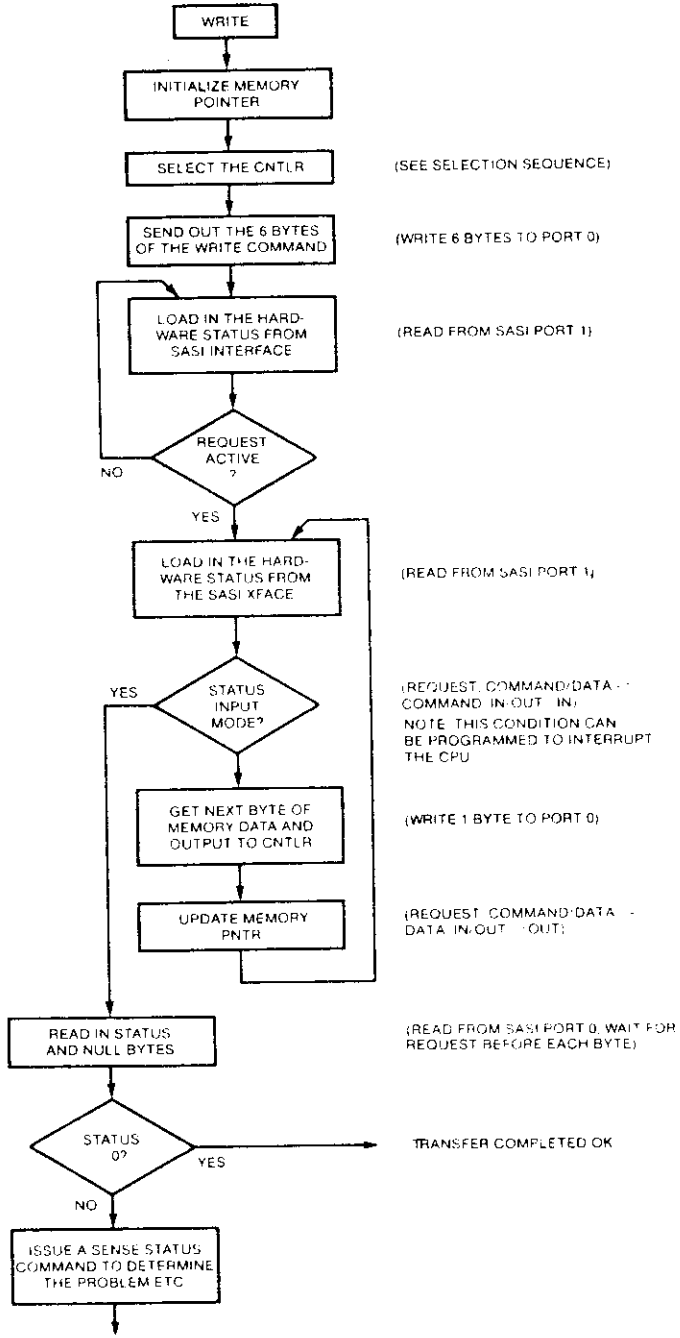


## READ USING NON-INTERRUPT I/O






## WRITE USING NON-INTERRUPT I/O



## Z80 PROGRAMMING EXAMPLE TO INTERFACE TO THE XEBEC S1410

THIS IS A STAND ALONE PROGRAM WRITTEN IN Z80 ASSEMBLY LANGUAGE WHICH SENDS COMMANDS TO THE XEBEC S1410 CONTROLLER AND 5¼" HARD DISK IN THE FOLLOWING ORDER:

1) RESET THE CONTROLLER
2) RECALIBRATE THE DRIVE
3) FORMAT THE DRIVE
4) WRITE ONE SECTOR FROM WRITEBUF
5) READ THE SAME SECTOR INTO READBUF

IF THE PROGRAM COMPLETES PROPERLY, THE READ BUFFER (READBUF) AND THE WRITE BUFFER (WRITEBUF) SHOULD BE THE SAME. NORMAL COMPLETION IS AT LABEL TEST6.

### OUTPUT AND INPUT PORTS

```
LABEL     OPCD  OPERAND  COMMENT
WPORT0    EQU   000H     SASI WRITE PORT 0 - WRITE DATA
WPORT1    EQU   001H     SASI WRITE PORT 1 - SOFTWARE RESET
WPORT2    EQU   002H     SASI WRITE PORT 2 - CNTLR SELECT
WPORT3    EQU   003H     SASI WRITE PORT 3 - NOT USED
RPORT0    EQU   000H     SASI READ PORT 0 - READ DATA
RPORT1    EQU   001H     SASI READ PORT 1 - READ STATUS
RPORT2    EQU   002H     SASI READ PORT 2 - NOT USED
RPORT3    EQU   003H     SASI READ PORT 3 - NOT USED
```

### VARIOUS EQUATES

```
REQBIT    EQU   000H     REQUEST LINE BIT POSITION
REQMASK   EQU   001H     REQUEST MASK FOR BIT TEST
BUSYBIT   EQU   001H     BUSY LINE BIT POSITION
BUSYMASK  EQU   002H     BUSY MASK FOR BIT TEST
MSGBIT    EQU   002H     MESSAGE LINE BIT POSITION
MSGMASK   EQU   004H     MESSAGE MASK FOR BIT TEST
CDBIT     EQU   003H     COMMAND/DATA BIT POSITION
CDMASK    EQU   008H     COMMAND/DATA BIT POSITION TEST
IOBIT     EQU   004H     INPUT/OUTPUT BIT POSITION
IOMASK    EQU   010H     INPUT/OUTPUT BIT MASK
```

### CONTROLLER COMMAND EQUATES

```
DRVREADY  EQU   000H     TEST DRIVE READY COMMAND
FORMAT    EQU   004H     FORMAT COMMAND CODE
READ      EQU   008H     READ COMMAND CODE
WRITE     EQU   00AH     WRITE COMMAND CODE
SENSE     EQU   003H     STATUS SENSE COMMAND CODE
INITL     EQU   00CH     INITIALIZE DISK SIZE COMMAND
SEEK      EQU   00BH     SEEK COMMAND CODE
RECAL     EQU   001H     RECALIBRATE COMMAND CODE
RAMDIAG   EQU   0E0H     RAM DIAGNOSTIC COMMAND CODE
ERROR     EQU   002H     TEST FOR AN ERROR
```



```
LABEL     OPCD  OPERAND          COMMENT
WRITEBUF  DS    256              WRITE BUFFER
READBUF   DS    256              READ BUFFER
STACK     DS    020H             CALL STACK
STACKTOP  EQU   $                TOP OF STACK
TASK      DB    0,0,0,0,1,80,F,0,0,0,0
```

### RESET

FIRST RESET THE CONTROLLER

```
        OUT     (WPORT1),A      SEND OUT A RESET PULSE
```

### TEST FOR DRIVE READY

TEST TO SEE THAT THE DRIVE IS UP AND READY AFTER CONTROLLER HAS BEEN SELECTED.

```
TEST1   EQU     $
        LD      SP, STACKTOP    SET THE STACK POINTER
        CALL    SELCNTLR        SELECT THE CONTROLLER
        LD      A, DRVREADY     DRIVE READY COMMAND
        CALL    TASKOUT         SEND OUT THE COMMAND
        CALL    GETSTAT         GET THE COMPLETION STATUS
        JP      Z, TEST2        TEST FOR PROPER COMPLETION
        RST     038H            ERROR COMPLETION
```

### RECALIBRATE

SEND OUT A RECALIBRATE TO THE CONTROLLER

```
TEST2   EQU     $
        CALL    SELCNTLR        SELECT THE CONTROLLER
        LD      A,RECAL         RECALIBRATE COMMAND CODE
        CALL    TASKOUT         SEND COMMAND TO CNTLR
        CALL    GETSTAT         GET COMPLETION STATUS
        JP      Z, TEST3        TEST FOR ERROR COMPLETION
        RST     038H            ERROR ON COMPLETION
```

### FORMAT

SEND A FORMAT COMMAND TO THE CONTROLLER

```
TEST3   EQU     $
        CALL    SELCNTLR        SELECT THE CONTROLLER
        LD      A,FORMAT        FORMAT COMMAND CODE
        CALL    TASKOUT         SEND COMMAND TO CONTROLLER
        CALL    GETSTAT         GET COMPLETION STATUS
        JP      Z,TEST4         TEST COMPLETION STATUS
        RST     038H            NON ZERO IS AN ERROR
```

| LABEL | OPCD | OPERAND | COMMENT |
|---|---|---|---|

## WRITE

WRITE OUT A SECTOR TO THE DISK

```
TEST4     EQU   $
          CALL  SELCNTLR      SELECT THE CONTROLLER
          LD    A,WRITE       WRITE COMMAND CODE
          CALL  TASKOUT       SEND TASK TO CONTROLLER
          LD    HL, WRITEBUF  POINT TO THE WRITE BUFFER
          CALL  REQWAIT       WAIT FOR CNTLR REQUEST
          IN

WRITE1    EQU   $
          IN    A,(RPORT1)    GET CNTLR STATUS LINES
          AND   CDMASK        TEST FOR COMMAND MODE
          JP    NZ,WRITE2     IF CMND. TRANSFER IS DONE
          LD    A,(HL)        GET A DATA BYTE
          OUT   (WPORT0),A    SEND DATA TO CONTROLLER
          INC   HL            BUMP THE BUFFER POINTER
          JP    WRITE1        MORE TO GO, LOOP
          IN

WRITE2    EQU   $
          CALL  GETSTAT       GET TRANSFER STATUS
          JP    Z,TEST5       TEST COMPLETION STATUS
          RST   038H          IF STATUS # 0, ERROR
```

## READ

READ A SECTOR FROM THE DISK

```
TEST5     EQU   $
          CALL  SELCNTLR      SELECT THE CONTROLLER
          LD    A,READ        READ COMMAND CODE
          CALL  TASKOUT       SEND COMMAND TO CONTROLLER
          LD    HL,READ BUF   POINT TO READ BUFFER
          CALL  REQWAIT       WAIT FOR REQUEST FROM CNTLR

READ1     EQU   $
          IN    A,(RPORT1)    GET CNTLR STATUS LINES
          AND   CDMASK        TEST FOR CMND MODE FROM CNTLR
          JP    NZ,READ2      IF ON, END OF TRANSFER
          IN    A,(RPORT0)    READ IN THE DISK DATA
          LD    (HL),A        SAVE IT IN THE BUFFER
          INC   HL            BUMP THE BUFFER POINTER
          JP    READ1         LOOP UNTIL 256 TRANSFERRED

READ2     EQU   $
          CALL  GETSTAT       GET COMPLETION STATUS
          JP    Z,TEST6       CONTINUE IF NO ERROR
          RST   038H          ERROR, STOP
          IN

TEST6     EQU   $
          RST   038H
```



| LABEL | OPCD | OPERAND | COMMENT |
|---|---|---|---|

## SELCNTLR

THIS SUBROUTINE SELECTS THE DEFAULT CONTROLLER

```
SELCNTLR  EQU   $
          IN    A,(RPORT1)    READ STATUS PORT
          AND   BUSYMASK      MASK BUSY BIT
          JP    NZ,SELCNTLR   JUMP, IF BUSY
          LD    A,1           CNTLR DEFAULT SELECT CODE
          OUT   (WPORT0),A    SEND IT TO TRANSPARENT LATCH
          OUT   (WPORT2),A    GENERATE A SELECT STROBE

SEL1      EQU   $
          IN    A,(RPORT1)    GET CNTLR RESPONSE
          AND   BUSYMASK      ISOLATE THE BUSY MASK
          JP    Z,SEL1        WAIT FOR CNTLR BUSY
          RET                 BUSY HAS ARRIVED, EXIT
```

## TASKOUT

THIS SUBROUTINE SENDS OUT THE COMMAND CONTAINED IN A REGISTER TO THE DISK CONTROLLER.

```
TASKOUT   EQU   $
          LD    HL, TASK      POINT TO TASK CONTROL BLOCK
          LD    (HL),A        SAVE THE COMMAND
          LD    B,6           SET UP A BYTE COUNTER
          CALL  REQWAIT       WAIT FOR CONTROLLER REQUEST
          IN

TASK1     EQU   $
          LD    A,(HL)        GET A COMMAND BYTE
          OUT   (WPORT0),A    SEND IT TO THE CONTOLLER
          INC   HL            BUMP THE TASK POINTER
          DEC   B             DECREMENT THE BYTE COUNT
          JP    NZ,TASK1      WAIT UNTIL ALL ARE OUTPUT
          RET
```

## GETSTAT

THIS SUBROUTINE RETRIEVES THE STATUS BYTE AND THE NULL BYTE FROM THE DISK CONTROLLER AT THE END OF A COMMAND. THE A REGISTER IS RETURNED WITH A NON-ZERO VALUE IF ERROR HAS BEEN DETECTED.

```
GETSTAT   EQU   $
          CALL  REQWAIT       WAIT FOR REQUEST
          IN    A,(RPORT0)    READ IN THE STATUS BYTE
          LD    D,A           SAVE STATUS TEMPORARILY
          CALL  REQWAIT       WAIT FOR SECOND BYTE
          IN    A,(RPORT0)    GET THE NULL BYTE
          LD    A,D           RESTORE STATUS TO A
          AND   ERROR         ISOLATE THE ERROR BIT
          RET
```

| LABEL | OPCD | OPERAND | COMMENT |
|---|---|---|---|

**REQWAIT**

THIS SUBROUTINE WAITS FOR THE REQUEST LINE TO BECOME ACTIVE FROM THE DISK CONTROLLER.

```
REQWAIT   EQU   $
          IN    A,(RPORT1)     GET CNTLR STATUS BITS
          AND   REQMASK        ISOLATE THE CONTROLLER
          JP    Z,REQWAIT      REQUEST AND WAIT FOR IT
          RET
          END
```

## Z-80 PROCESSOR INTERFACE TO THE XEBEC S1410 CONTROLLER

Z-80 SIGNAL DEFINITIONS

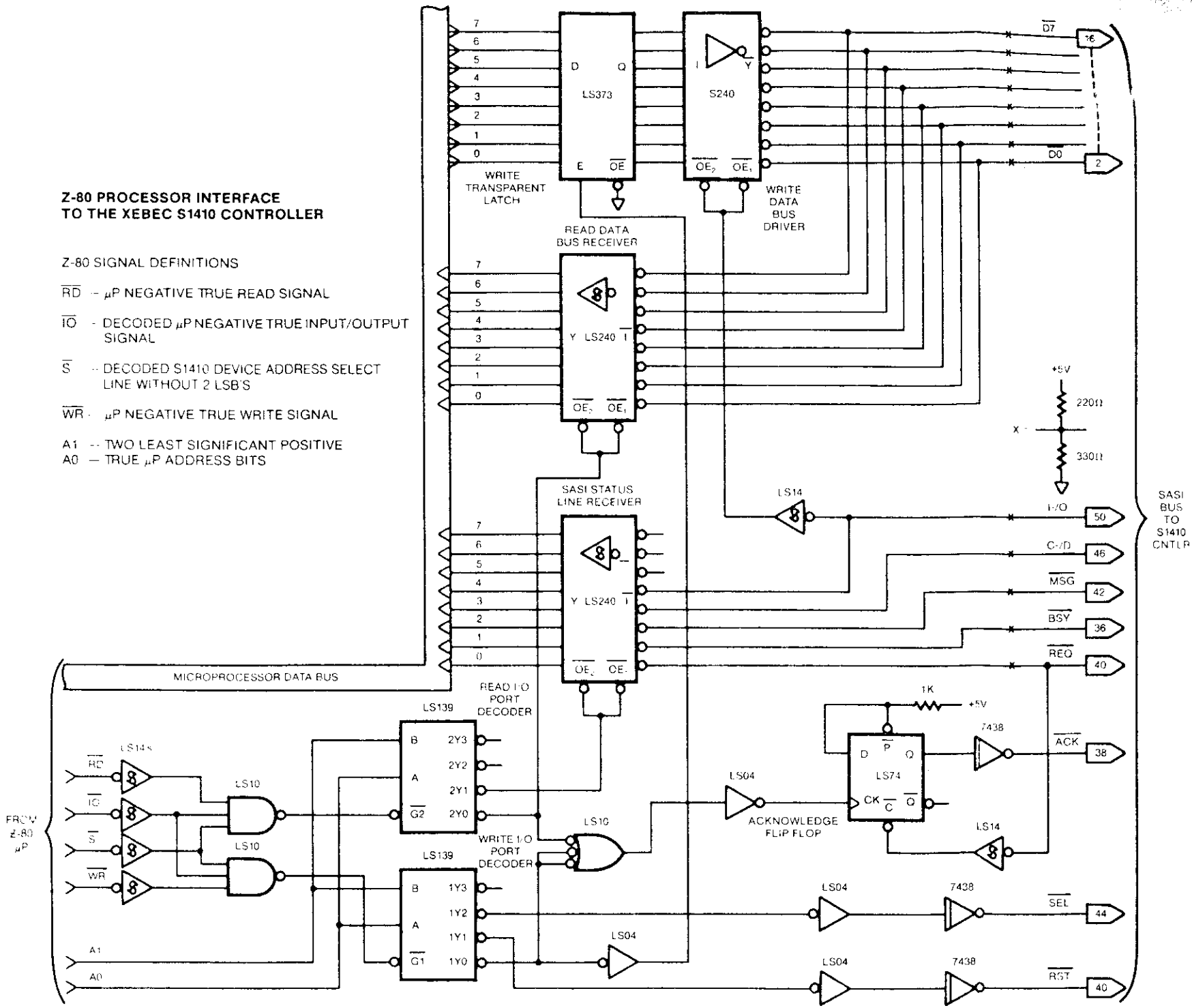
$\overline{RD}$ – μP NEGATIVE TRUE READ SIGNAL

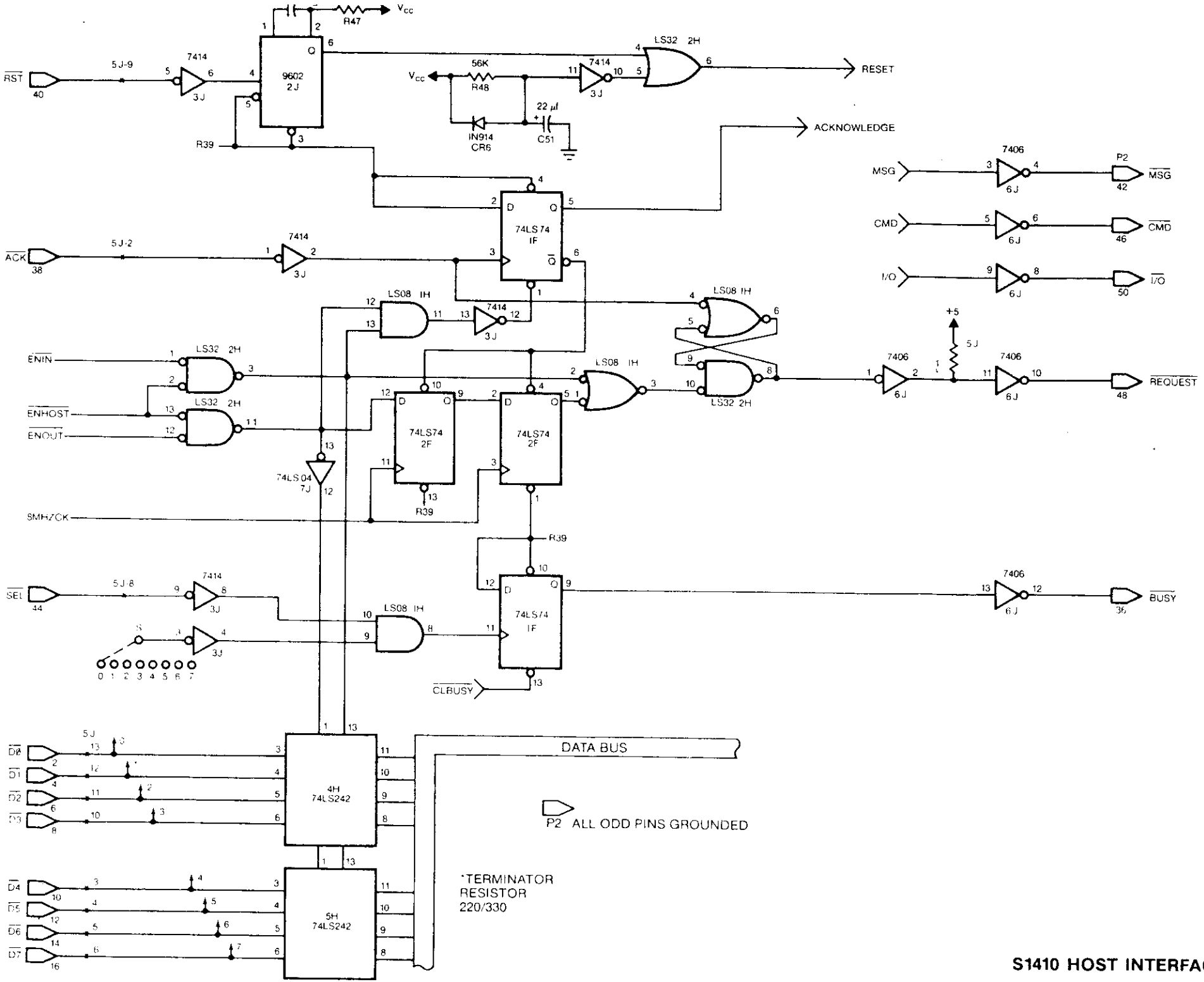
$\overline{IO}$ – DECODED μP NEGATIVE TRUE INPUT/OUTPUT SIGNAL

$\overline{S}$ – DECODED S1410 DEVICE ADDRESS SELECT LINE WITHOUT 2 LSB'S

$\overline{WR}$ – μP NEGATIVE TRUE WRITE SIGNAL

A1 – TWO LEAST SIGNIFICANT POSITIVE
A0 – TRUE μP ADDRESS BITS






**S1410 HOST INTERFACE**

# XEBEC S1410
# OWNER'S MANUAL
# SUPPLEMENT

AUGUST, 1982

# S1410 REV E FIRMWARE

There are major changes on the S1410 Rev E firmware from Rev. D. All changes are downward compatible with software for Rev. D.

## A. NEW COMMANDS ADDED

### 1. Format Alternate Track (OPCODE 'OE' Hex)

Format Alternate Track will format the header fields of the "Bad Track" with the alternate track information (assigned by the host). The alternate track is formatted to identify it as an alternate. The command bytes for Format Alternate Track are:

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| Byte 1 | 0 | d | d | LOGICAL ADDRESS HI | | | | |
| Byte 2 | LOGICAL ADDRESS MIDDLE | | | | | | | |
| Byte 3 | LOGICAL ADDRESS LO | | | | | | | |
| Byte 4 | 0 | 0 | 0 | INTERLEAVE | | | | |
| Byte 5 | r | r | F | 0 | s | s | s | s |

d = drive, 0 or 1 s = step option, r = retries.

The logical address in the command bytes point to the "Bad Track". Sector address is ignored, defaulting to sector 0.

The interleave byte (4) is programmed the same as in the format command, and is used on the alternate track.

If Bit 5 of Control Byte (5) is set, the data in the existing sector buffer is used to fill the data field. If not set, the data field is written with hex 6C.

After issuing the command the controller will ask for 3 bytes. These bytes point to the host assigned alternate logical address. Again sector address is ignored.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | LOGICAL ADDR HI | | | | |
| Byte 1 | LOGICAL ADDRESS MIDDLE | | | | | | | |
| Byte 2 | LOGICAL ADDRESS LO | | | | | | | |

After receiving the command and the assigned alternate, the controller does the following:

a) Seeks to "Alternate Assigned Track" and verifies it is not already an assigned alternate track, or flagged bad track.
   - If the track has already been assigned as an alternate or is flagged "BAD", then error code ID Hex is given, and the command is aborted. This usually implies that the host is attempting to assign two (2) bad tracks to the same alternate track.

b) Formats the track as an assigned alternate track.

c) Seeks to the "Bad Track" and formats the header as a spare track pointing to the assigned alternate.

**NOTE:** Data fields on both the bad track and alternate track are destroyed.

Using the Format Alternate Track Command

a) The controller must be initialized to include the alternate tracks cylinder and head ranges.

b) With alternate tracks, the entire disk is not available to the system. Generally the disk space is fixed in the system software, which leaves spare tracks at the inner cylinder of the disk, which can be assigned as alternates when needed.
   The number of spare tracks is dependent on drive size and number of defects allowed by the drive manufacturer. Generally this is 1 spare track for each 50 to 100 tracks.

c) Procedure for use is:
   1) Format entire disk, including spare tracks.
   2) Verify disk.
   3) For each media defect, assign an alternate track.
   4) Alternate tracks should also be assigned for drive manufacturer list of defects.

d) In system operation, the alternate tracks are invisible to the host. The controller will automatically seek to the assigned alternate track when an access is made to a flagged defective track. "Consecutive" accesses to a flag track does not result in reseeking to the alternate track. The controller will maintain position on the alternate track.

e) Direct access (seeking to, or attempted data transfer) to an alternate track results in an error code 'IC' Hex, and no data transfer takes place.

**2. Write Sector Buffer (OPCODE 'OF' Hex)**

Used to fill the sector buffer with a host given data pattern. No transfer of data takes place between the drive and the controller.



| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| Byte 1 | / | / | / | / | / | / | / | / |
| Byte 2 | / | / | / | / | / | / | / | / |
| Byte 3 | / | / | / | / | / | / | / | / |
| Byte 4 | / | / | / | / | / | / | / | / |
| Byte 5 | / | / | / | / | / | / | / | / |

This command accepts 256 or 512 bytes (depending on sector size jumper) of data, and stores it in the sector buffer.

**3. Read Sector Buffer (OPCODE '10' Hex)**

This command sends 256 or 512 bytes of data (depending on sector size jumper) to the host from the sector buffer.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Byte 1 | / | / | / | / | / | / | / | / |
| Byte 2 | / | / | / | / | / | / | / | / |
| Byte 3 | / | / | / | / | / | / | / | / |
| Byte 4 | / | / | / | / | / | / | / | / |
| Byte 5 | / | / | / | / | / | / | / | / |

## B. CHANGES TO FORMAT DRIVE COMMAND

If Bit 5 of control byte 5 of the command block is set on the format command (OPCODE 04), the sector buffer will be used as the data pattern written on the disk data fields.

The Write Sector Buffer command can be issued before the format command to initialize the sector buffer.

## C. CHANGE TO FORMAT TRACK COMMAND

The Format Track Command will reformat the track, eliminating all references to bad and alternate tracks. Also if Bit 5 of control byte 5 of the command block is set, then the sector buffer will be used as the data pattern in the data field.

## D. EXPANDED FAST STEP OPTIONS

The fast seek steps supported have been increased to include many of the popular drives. The new step option selection is downward compatible with previous step options supported.

The step options are encoded in control byte 5 of the command descriptor. The encoding is done with bits 0 thru 3 as follows:

| DESCRIPTION | BITS 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|
| Default 3 msec step rate | 0 | 0 | 0 | 0 | |
| Seagate ST506 (MLC2) | 0 | 0 | 0 | 1 | |
| Tandon fast step | 0 | 0 | 1 | 0 | |
| Texas Instrument fast step | 0 | 0 | 1 | 1 | |
| 200 usec buffered step | 0 | 1 | 0 | 0 | |
| 70 usec buffered step | 0 | 1 | 0 | 1 | |
| 30 usec buffered step | 0 | 1 | 1 | 0 | |
| 15 usec buffered step | 0 | 1 | 1 | 1 | 7 |
| Olivetti 2msec step (561) | 1 | 0 | 0 | 0 | |
| Olivetti (562) fast step (1.1 msec typical) | 1 | 0 | 0 | 1 | |
| Spare (for future use) | 1 | 0 | 1 | 1 | thru |
| | 1 | 1 | 1 | 1 | |

Refer to the drive manufacturer manual in configuring the drive for fast, or buffer step options. In cases where the drive is hardware configured for fast step, all commands which require seek option selection must use the fast step option for that drive.

## E. DESCRIPTION OF OVERLAP SEEKS WITH BUFFERED STEP DRIVES

For drives employing buffered seeks, seek commands can be overlapped. After the controller issues a seek to the drive, it returns with a completion status, not waiting for the drive to complete the seek. If the return status shows no error, then the seek was issued correctly. If there is an error, then the seek was not issued. After transferring the status, another command can be issued to either drive. If a new command is received for a drive with an outstanding seek, then the controller will wait, with Busy active, for the seek to complete before executing the new command (Except Test Drive Ready Command). There is no timeout condition in the controller, waiting for the buffered step seeks to complete.

The Test Drive Ready command can be used with overlapped seeks to determine when a drive has completed seeking before issuing the next command. If the drive is still seeking, the status byte at the end of the command will indicate an error, and the sense status will indicate "drive still seeking" (type 0 error, code 8). A sequence of Test Drive Ready commands can thus be used to determine when the drive is ready for the next command.

## F. NEW ERROR CODES

**NOTE:** The address Valid Bit (bit 7) may or may not be set, and is not included here for clarity.

| Error Code (Hex) | Description |
|---|---|
| 1C | Illegal (direct) access to an alternate track. |
| 1D | On a format alternate track command, the requested alternate track has already been assigned as an alternate, or is flagged as a bad track. |
| 1E | When the controller attempted to access an alternate track from a spared track, the alternate track was not flagged as an alternate. |
| 1F | On a format alternate track command, the bad track equalled the alternate track. |

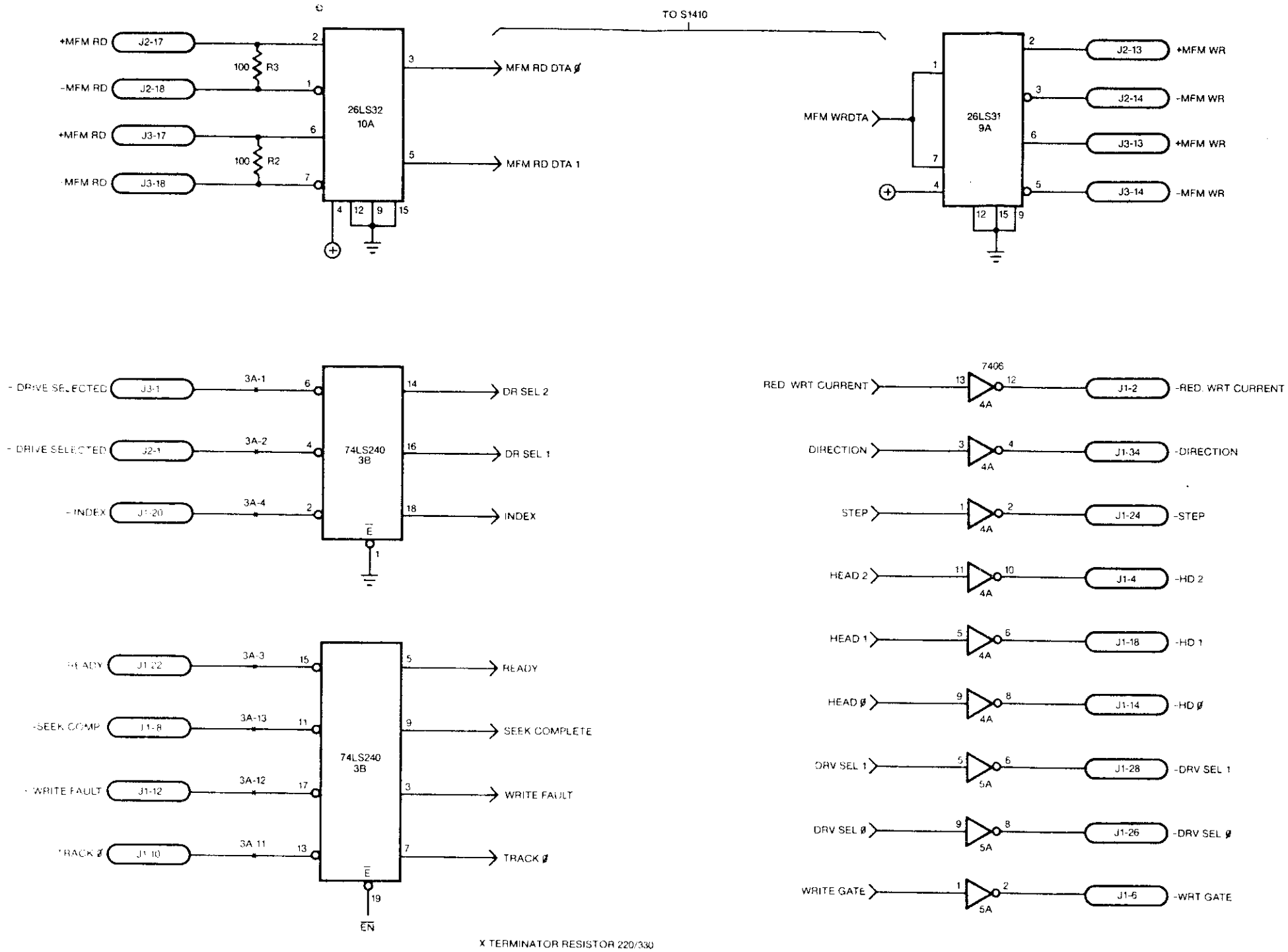
# TIMING CLARIFICATIONS

## A. CONTROLLER SELECTION

The SELECTION signal should NEVER be activated (SEL - =0) within 1 microsecond of the deactivation of the BUSY signal (BUSY - =1) or within 10 microseconds after a RESET; if this timing is violated, the controller may not be able to accept the selection sequence.

## B. OTHER TIMING CONSIDERATIONS

| FROM | TO | TIME |
|---|---|---|
| SEL | REQ | 100 microseconds |
| RESET OR POWER ON | CONTROLLER READY | 2 milliseconds |
| WRITE COMMAND | 1ST DATA REQUEST | 200 microseconds |
| MESSAGE BYTE | END OF BUSY | 10 microseconds |




X TERMINATOR RESISTOR 220/330

**S1410 DRIVE INTERFACE**